DOCUMENT RESUME

ED 024 410 LI 001 088

By- Spigai, Frances. And Others

A Pilot - An On-Line Library Acquisition System.

Oregon State Univ., Corvallis. Computer Center.
Report No- CC-68-40
Pub Date Jan 68
Note- 117p.
EDRS Price MF-$0.50 HC-$5.95
Descriptors- •Automation, Computers, Display Systems, Information Processing, Information Storage, •Library Acquisition, •Man Machine Systems, •Programing, Purchasing, Search Strategies, Systems Development, •University Libraries

This report presents documentation of files, of file organization, and of 42 of the modular program subroutines used in a pilot project devised by the Oregon State University (OSU) Computer Center to simulate procedures in the Acquisitions Department of the OSU Library. A total of 224 bibliographically verified requests for monographs, 30 vender names and addresses, and 62 campus departments having library fund allocations comprise the three files constituting the main data base for the pilot project. Fixed field formats are used for bibliographic input, the vendor file uses a variable field input. Searching is on both fixed and variable lengths and the initial methods of accessing elements in the files are through table lookup, sequential match, and algorithmic search and retrieval strategies. The modes of interaction among the computer configuration, the programming system and the user in the project are on-line, real-time, time-sharing, and conversational. Documentation of programs in this report assumes the use of a cathode ray tube terminal as the input/output device; however, these same programs have since been modified for use on the teletypewriter. An area of programming not included is the algorithmic search used for locating the main entry. This report documents work done from May 1967 through January, 1968. (Author/CM)

EDO 24410 8c-68-40

LE 001088

ERIC/CLIS
NOV 22 '68
UNIVERSITY OF MINNESOTA

A Pilot - An On-Line Library Acquisition System

Prepared by
Frances Spigai
Mary Taylor

Project Director
Michael A. Jennings

January, 1968

LE 001088

OSU
COMPUTER CENTER
Oregon State University
Corvallis, Oregon 97331

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

A PILOT

AN ON-LINE LIBRARY ACQUISITION

SYSTEM

January, 1968

Prepared by:

Frances Spigai
Mary Taylor

Programming Support:

Thomas Brantner
Mary Taylor
Neil Tokerud

Project Director:

Michael A. Jennings

Developed by
Oregon State University
Computer Center
Corvallis, Oregon

cc-68-40

TABLE OF CONTENTS

TEXT	Pages
ABSTRACT	i
INTRODUCTION	1
INPUT TO SYSTEM	2
Search Capabilities	2
1. Main-entry	3
2. Library of Congress classification number	3
3. Order number	3
PROCESSING	4
1. Equipment	4
2. Programming	4
3. Environmental Operating Modes	5
4. File Maintenance	5
OUTPUT OF SYSTEM	7
SPECIAL FEATURES	8
FUTURE CAPABILITIES	8
APPENDICES	
A. EXPLANATIONS AND SAMPLES OF THE WORKSHEETS TO BE USED FOR THE OUTSTANDING ORDER FILE RECORDS AND THE VENDOR FILE RECORDS	11
B. OREGON STATE UNIVERSITY COMPUTER CENTER CONFIGURATION (1967)	15
C. EQUIPMENT SET-UP PROCEDURES FOR THE PILOT	16
D. GENERAL PROGRAM DESCRIPTION, FLOWCHARTS AND THE REPRESENTATIONS	17
E. LISTINGS OF PILOT SUBROUTINES	25
F. FLOWCHARTS OF PILOT SUBROUTINES	27
G. LISTING AND DEFINITIONS OF VARIABLE NAMES USED	102

ABSTRACT

This report presents documentation of files, file organization and of 42 of the modular program subroutines used in a pilot project devised by the Oregon State University Computer Center to simulate procedures in the Acquisitions Department of the OSU Library. A total of 224 bibliographically verified requests for monographs, 30 vendor names and addresses, and 62 campus departments having library fund allocations comprise the three files constituting the main data base for the pilot project. Fixed field formats are used for bibliographic input; the vendor file uses a variable field input. Searching is on fixed and variable lengths and the initial methods of accessing elements in the files are through table lookup, sequential match, and algorithmic search and retrieval strategies.

The modes of interaction among the computer configuration, the programming system and the user in the project are on-line, real-time, time-sharing and conversational. Programming is in CDC FORTRAN and the hardware used is a CDC 3300 computer with 65K storage, an 854 diskpak (eight million alphacharacter storage), a CDC 604 tape drive, a 501 line printer, a CDC 405 card reader and a CDC 210 cathode ray tube (CRT) terminal.

The report documents work done from May, 1967 through January, 1968. Documentation of programs in this report assumes the use of a CRT as the I/O device; however, these same programs have since been modified for use on the teletype - primarily as a cheap method of program debugging. Current

work on more sophisticated table-based search methods should accomodate additional features desired for a library data base requiring variable field formats. An area of programming not included in this report is the algorithmic search used for locating main entry.

FS 10/68

INTRODUCTION

The objective of the pilot project is to devise an on-line computer system model which simulates the ordering procedures for the Oregon State University Library's Acquisitions Department. The model utilizes a cathode ray tube (CRT) display as a terminal device. The specific procedures which are available through the computer system model are: 1) purchase-order generation, 2) main entry search of the outstanding order file, 3) Library of Congress classification number search, 4) unique order number search, and 5) complete maintenance of the four files which currently make up the order system.

To handle the diverse requirements of a library acquisitions system, a series of integrated files have been structured. Four files are presently being used:

(FIXED FIELD) 1) Purchase File. This file contains records of requests (vs. orders) for monographs. The records are in sequential order; their arrangement is a function of when they are received and the processing each must undergo. This would currently take care of all the files of requests which have been verified, but not ordered. (e.g. hold for money file; those batched for sending to a particular vendor). When these are assigned an order number, a purchase order is selectively generated and the record is automatically transferred to the outstanding order file.

(FIXED FIELD) 2) Outstanding Order File. This file contains records of all orders which have been sent out, but have

not been received. They are arranged alphabetically by main entry, or arranged in a rough alphabetical order by main entry by an algorithm which uses letters in the main entry to calculate a location.

(VARIABLE FIELD) 3) Vendor File. A list of the 30 vendors (and their mailing addresses) used most frequently by the Acquisitions Department forms this file. Records are arranged in sequential order as they are input; the internal storage location corresponds to the code each vendor is assigned.

(FIXED FIELD) 4) Department File. This consists of a list of departments, arranged alphabetically. The department location in the file corresponds to the code numbers assigned to each of them.

The worksheet for converting records to the purchase file and the outstanding order file is shown in Appendix A; the worksheet for the vendor file is also shown. The departmental file contains names of the 62 departments on campus which have library fund allocations.

INPUT TO SYSTEM

Search Capabilities

Currently, search capabilities exist for three parameters: main-entry, Library of Congress classification number, and order number.

1. Main-entry searches have the greatest flexibility, since this is the most common way for the outstanding order file to be accessed. Access is through one of two methods 1) table look-up and 2) algorithmic calculation.

Programming work is now being studied which will take advantage of variable length fields and records and will include a multi-level index search scheme utilizing pointers (not too dissimilar to search systems used by Project INTREX at MIT and SPIRES at Stanford University). Search on variable length (e.g. Jones, Arthur C., or Jones, A. C., or Jones, A., or Jones) and capabilities to step through the file record by record in either direction (A → Z or Z → A) have been programmed. If the entry is incomplete (e.g. if one has only an author's last name and one initial Jones, A.) it can be searched and subsequent records can be stepped through by using the step forward capabilities built into the system. If the entry for which one is searching is not contained in the file, a message with this information will appear on the display screen.

2. Library of Congress classification number search is a straightforward matching process. The desired LC number is typed, after which it is located through a sequential search and displayed. Since this search procedure is not economical for a large data base, a table-based look up is envisioned. If the LC number is not located, a message with this information will appear on the display screen.

3. Order number search is also a sequential matching process at this time, though, as with the LC number search, this procedure will be changed for a large data base. If the order number is not located, the user will be informed by a message on the display screen.

PROCESSING

1. Equipment

a. A CDC 3300 digital computer is used as the central processing unit for the system. It has a memory capacity of 65K. It is now operating under the OS-3 time-sharing executive developed by the OSU Computer Center. The programming languages available under OS-3 include two compilers (FORTRAN and ALGOL), OSCAR (the OSU - developed conversational language), and the CDC 3300 assembly language COMPASS.

b. A CDC 854 disk with a memory capacity of eight million alphanumeric characters is used as a storage unit.

c. Magnetic tape back-up for all files.

d. CDC 210 CRT. The on-line input/output terminal used for the pilot system is a CDC 210 Display and Entry terminal. The screen has the capacity for 1000 alphanumeric characters. These characters are buffered from the disk through the CDC 3300 and onto the display tube screen. Data on the screen plus responses to the system determine what is buffered from the screen back into disk memory.

2. Programming

CDC FORTRAN is used as the main system programming language. Most of the CRT utility subroutines are written in COMPASS

(CDC assembly language). It is felt that use of the widely known programming language FORTRAN will partially solve future problems of turnover both in programming staff and computer equipment. The logic used for programming the library automation project is a modular one in that the programs are like a set of blocks, each block representing a subroutine. To expand the programming system, blocks are added. Each subroutine is independent of the total programming system so that if subroutines are added, deleted, or modified, the effects on the system as a whole are kept to a minimum.

3. Environmental operating modes (Conversational real-time mode)

The program is written to simulate a query-response conversational mode between the computer and the user. Actions taken by the system depend upon the response of the user to choices presented by the system on the display screen. Thus, in natural language (vs. programming languages), there is a real-time dialogue between the user and the computer allowing the user to operate the system with a minimum of technical knowledge.

4. File maintenance

a. Outstanding order file

1. Add New Record: New records are added at the end of the alphabetically arranged (by main-entry) records in the outstanding order file; thus, in searching, the computer searches through two sections. The file will be re-alphabetized (sorted) automatically in whatever time interval is designated (once a day, week, etc.).

2. Correct: Corrections, additions and changes to records already in the outstanding order file are made by locating the record and writing in new information or writing over old information when making corrections (a function of the CRT device).

3. Delete: Locate the record to be deleted; a YES response to the delete subroutine's question: "Is this the desired record?", will delete the record.

b. Purchase file

Additions of new records are made and stored in the order that they are put into the system. Corrections and additions to a record can be made at any time in a manner analogous to that used in the outstanding order file.

Records in the purchase file are automatically transferred by the system to the outstanding order file upon the printout of a purchase order. The routine for deletions in the purchase file is the same as that for the outstanding order file.

c. Vendor file

1. Additions to the vendor file are added on to the end of the file. Code numbers are equivalent to the location of the vendor internal programming array. Alphabetical listings of vendors and their codes are printed for Acquisitions Department personnel so that they can determine the code number of a vendor. The computer automatically assigns and displays the next code number available for a new vendor.

2. Corrections: Since records in the vendor file are of variable length, there are two methods of correction:

a. If the corrected record is the same length or less than the original record, the original record is written over and corrected in place.

b. If the corrected record is longer than the original record, the corrected version is added to the end of the vendor file and access is made possible from the old location code to the new location code through programming directions.

3. Deletions: The vendor record to be deleted is located and is deleted by the user's YES response to the system's question: "Is this the desired record?"

d. Departmental files

At the present time the departmental file consists of 62 departments arranged alphabetically. Their alphabetical position corresponds to their memory location which is equivalent to their code number. This file is small and due to University funding arrangements, normally changes only once a year. Therefore, no programming for deletions, additions, and corrections has been done, since it is felt that this can be taken care of most economically by generating a new file.

OUTPUT IS GIVEN IN TWO MEDIA

1. Screen display is available for all questions, answers and displays of a file record.
2. Hardcopy is available for any purchase order or outstanding

order. Work is now underway towards a hardcopy of any displayed response.

SPECIAL FEATURES

1. Vendor status

The system determines how many orders in the purchase file have been issued to a particular vendor. This capability helps in batching and sending orders to vendors.

2. Multiple-title main-entries

In the purchase file, where there is more than one title for a particular author, the first record for that author is displayed upon request. The system automatically searches for and displays subsequent titles by that author until the user writes STOP, or until the last record for that author is displayed. The last record will be labeled EOF (end-of-file).

FUTURE CAPABILITIES AND SUGGESTIONS

1. The programming for the pilot project is being modified for use on a teletype. Although the system will be programmed to work on a CRT terminal, on-line, time-shared debugging is much cheaper via teletype. Programs developed on a teletype can be converted for use on a CRT with little program modification.
2. On the main entry search in the outstanding order file, the first record for an entry with several records will display:

a. the number of records under that entry, or

b. READY (and search for the next record(s) until END OF FILE appears on the last or only record) so that step forward will not have to be used with every main-entry search.

3. The information that the user entered should be displayed at the same time as the NOFIND information statement ("Either the record is not on file or you used the incorrect format to enter information"), so that the user will know whether his input was in error.

4. Make the main-entry either variable field or a longer fixed field to assure a complete main-entry. Some drawbacks to the use of a variable field include: a) the need to place the main-entry at the end of each record, resulting in the whole record being buffered in for a search, which extends response time and increases costs; b) updating of the file using a variable length record means more intricate programming.

5. Information statements should make it clear that step forward/step backward capabilities do not exist for order number or LC number searches since the purchase and outstanding order files are organized by main-entry.

6. Search capability on series, which might include linking of records.

7. Main-entry search should be coupled with title search (for entries like Shakespeare and U. S. Department of

Interior) for more rapid access to the desired record.

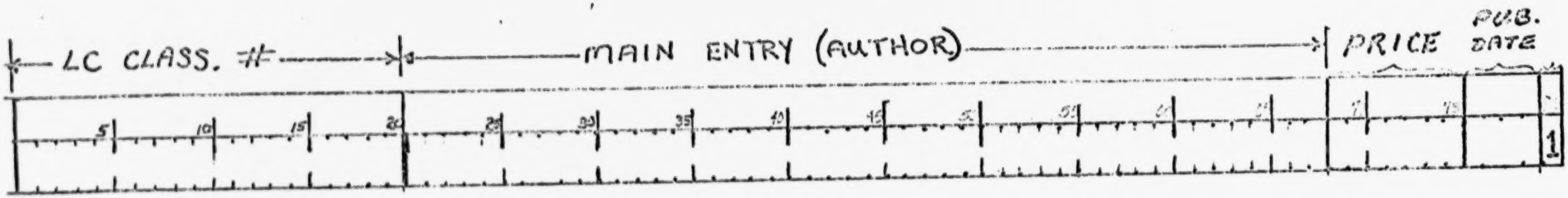
8. File security for updating (correction, deletion and addition) of outstanding order and purchase files.
9. Hardcopy printout for each Oregon State University department at convenient intervals (i.e. once a month, once a quarter, etc.) should give the following information: a) the status of books they have requested (on order, in process, etc.) b) budget status report.
10. Modify the worksheet and put the order number where the price is - that way the whole record would not have to be buffered in for the order search (just the first 75 characters).
11. Build in a procedure for automatic claiming (e.g. match the date of order, January 1 for example, with a January 1 card put in on January 30, for a 30 day claim period).
12. Management Information System for the Acquisitions Department. With certain programming plus information contained in files questions of the following type could be answered:
    a. How many orders in a certain time period (or at a certain time) have been placed with a book vendor?
    b. What is the average length of time it takes to order from a particular vendor?
    c. How many orders are requested during certain months, ordered during certain months, received during certain months?

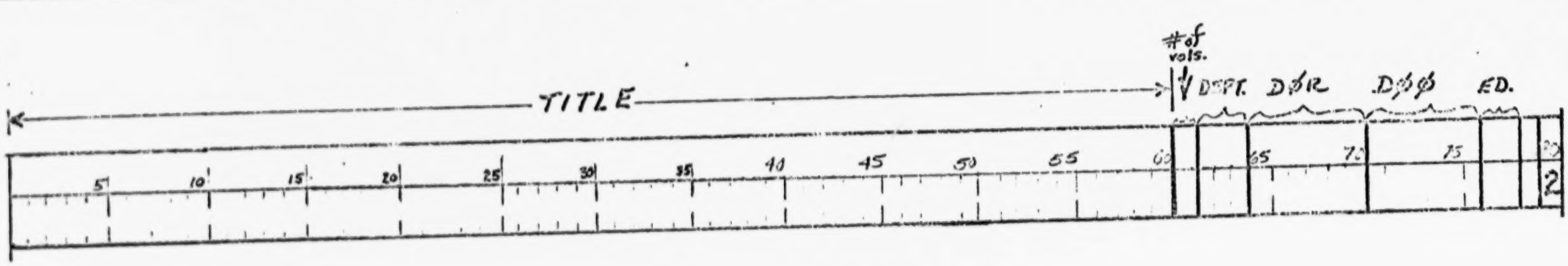
APPENDIX A

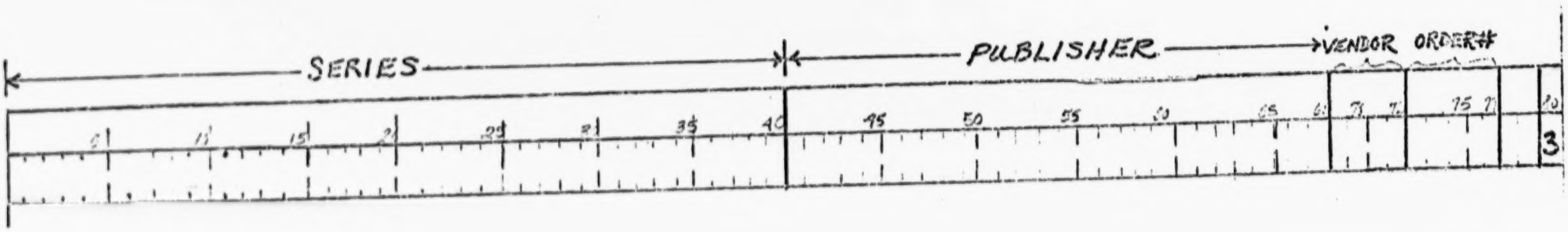
EXPLANATION OF FIXED FIELDS ON THE OUTSTANDING ORDER FILE WORKSHEET **

	Card Columns	Data Element
Card #1	1. 1-20	LC Classification Number
	2. 21-68	Main Entry **
	3. 69-75	Price
	4. 76-79	Publication Date
	5. 80	Punched-card sequence no. (1)
Card #2	1. 1-60	Title
	2. 61	Number of volumes
	3. 62-64	Department Code (e.g. 001= Agricultural Engineering)
	4. 65-70	Date of Request
	5. 71-76	Date of Order
	6. 77-78	Number of Edition
	7. 79	Reserved for Future Needs
	8. 80	Punched-card sequence no. (2)
Card #3	1. 1-40	Series
	2. 41-68	Publisher
	3. 69-72	Vendor
	4. 73-77	Order No.
	5. 78-79	Reserved for Future Needs
	6. 80	Punched-card sequence no. (3)

** Although input of the main entry is restricted to a fixed field, there is currently a working capability for searching on a variable length (e.g. Smith, John Allen, or Smith, John A, or Smith, J, or Smith, etc.) within a fixed field.


LC CLASS. #
MAIN ENTRY (AUTHOR)
PRICE
PUB. DATE
1


# of vols.
TITLE
DEPT. DØR
DØØ
ED.
2


SERIES
PUBLISHER
VENDOR ORDER#
3

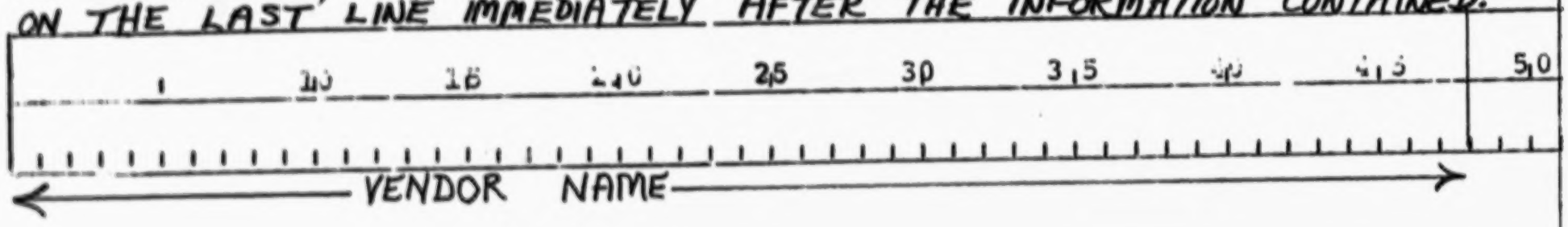
EXPLANATION OF VARIABLE FIELDS ON THE VENDOR NAME AND ADDRESS WORKSHEET

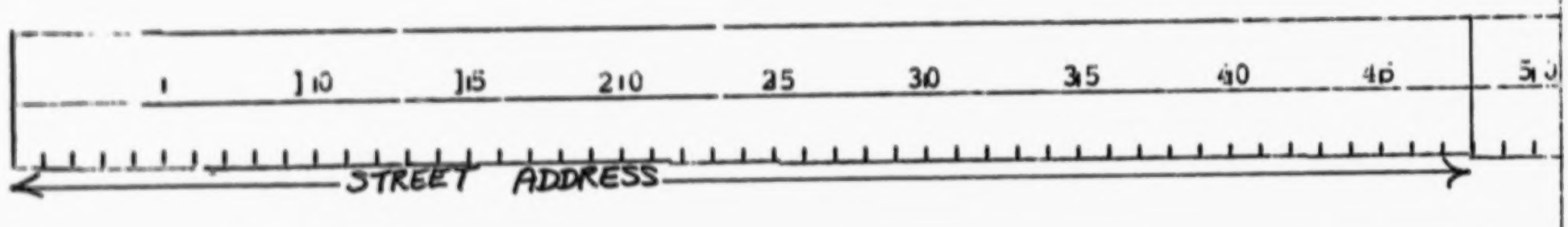
A maximum of 5, 48-column cards can be used for the vendor address. The end of each line of the address is flagged with a dollar sign ($), except for the last line. The end of the last line is flagged with an asterisk (*), denoting the end of that particular vendor record.

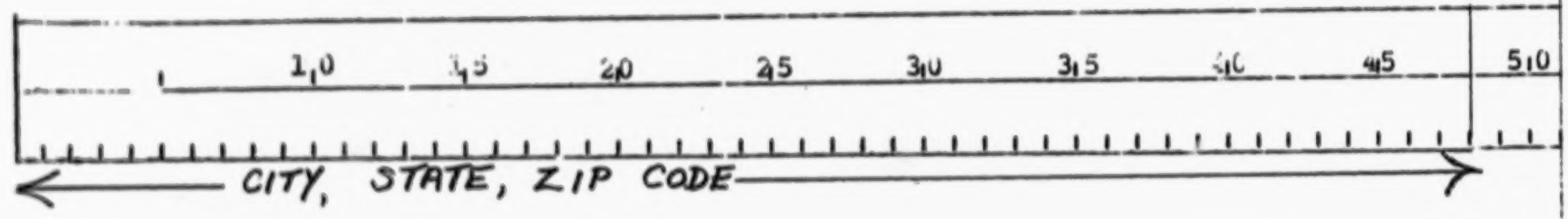
The use of variable fields for the vendor records permits a maximum packing of records and saving of space. A printout of the vendor record is substituted for the vendor code on the purchase order which is sent out.

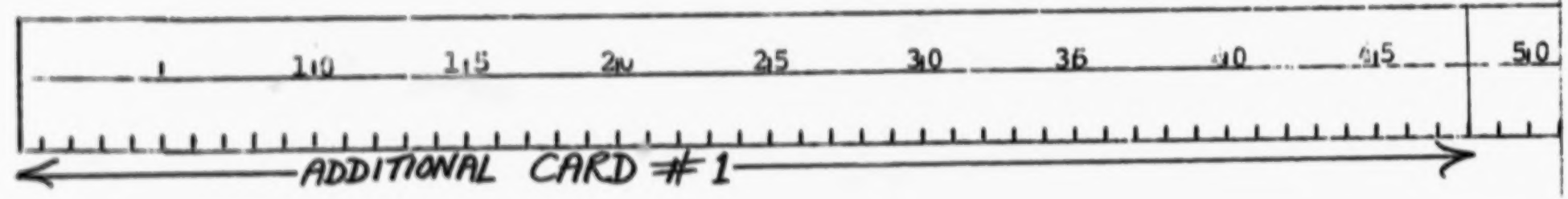
The vendor code is equivalent to the address location of the vendor record on the disk; therefore, it is not necessary to use additional storage space for code numbers, nor is it necessary to keep track of code-number assignments since these are assigned automatically by the computer.

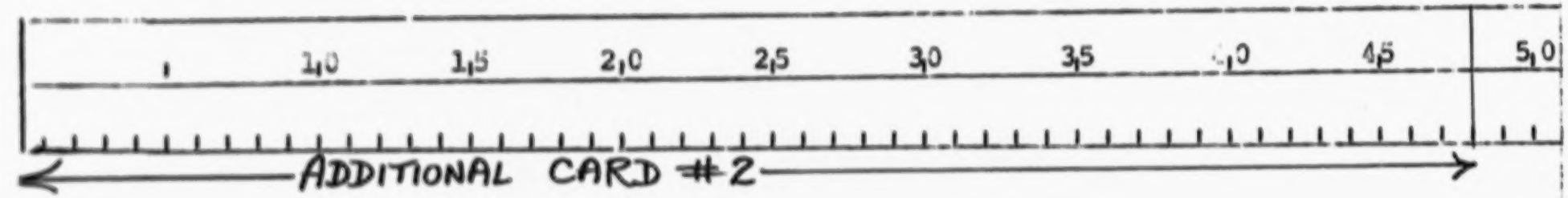
WRITE A DOLLAR SIGN ($) IMMEDIATELY AFTER THE INFORMATION CONTAINED ON EACH LINE, EXCEPT FOR THE LAST LINE. WRITE AN ASTERISK (*) ON THE LAST LINE IMMEDIATELY AFTER THE INFORMATION CONTAINED.











APPENDIX B

OREGON STATE UNIVERSITY

COMPUTER CENTER CONFIGURATION*
(1967)

CDC 3300 Computer
65K of core memory
6 I/O channels
Multiprogramming Time Sharing
Paging

Amount	Type of Device
6	CDC 604 tape drive (200,556,800 bpi)
3	CDC 854 disk drive
3	CDC 852 disk drive
1	CDC 501 line printer
1	CDC 505 line printer
2	CDC 3276 multiplexor (16 lines)
1	CDC 405 card reader
1	CDC 415 card punch
4	CDC 210 CRT
1	CDC 211 CRT + data set
30	KSR 33 & 35 teletypes

*The configuration has experienced several changes in the past nine months, the most notable being the addition of a CDC 814 mass storage disk.

APPENDIX C

EQUIPMENT SET-UP PROCEDURES

Turn on CRT
Master clear CRT
Clear CRT
Mount tape 1209 on unit 3
Mount disk 7629 (852/01) on free 852 disk drive
Load deck in card reader
Press "motor on" button on card reader
Press "ready" button on card reader
On console
    Press "stop" button
    Press "MC" button
    Press "autoload" button
    Press "select jump 6" button
    Console types: "date"
    Type in current date
    Console types: "inp"
    Press "finish" button
    Console types: "out"
    Type "16"
    Press "finish" button
    Console types: "run"
    Press "select jump 6" button
    Press "finish"

If program not accepted, reload deck in card reader, press "MI" button, type "/", press "MI" button.

APPENDIX D

Program DISPLAY

First the title page is displayed. After the SEND button is pushed, the choice of files is displayed. The user indicates which file he wishes to work with and either the OSOFILE (outstanding orders file), VENDFILE (vendor file), PFILE (purchase file) or SIGNOFF subroutine is called.

If the user chooses to work with the outstanding order file a set of choices is presented pertaining to this file. Capabilities and their respective subroutines are: 1) step forward through the file one record at a time (by calling RECORD), 2) step backward one record at a time (BACK), 3) search for a record by its main entry (SAUTH), 4) search for a record by its LC classification number (SLC), 5) search for a record by its order number (SORDER), 6) move to a different file (FILECH which brings the user back to the original set of choices), 7) correct a record (CORECT), 8) delete a record from the file (DELETE), and 9) add a record directly into the outstanding order file (OSONEW).

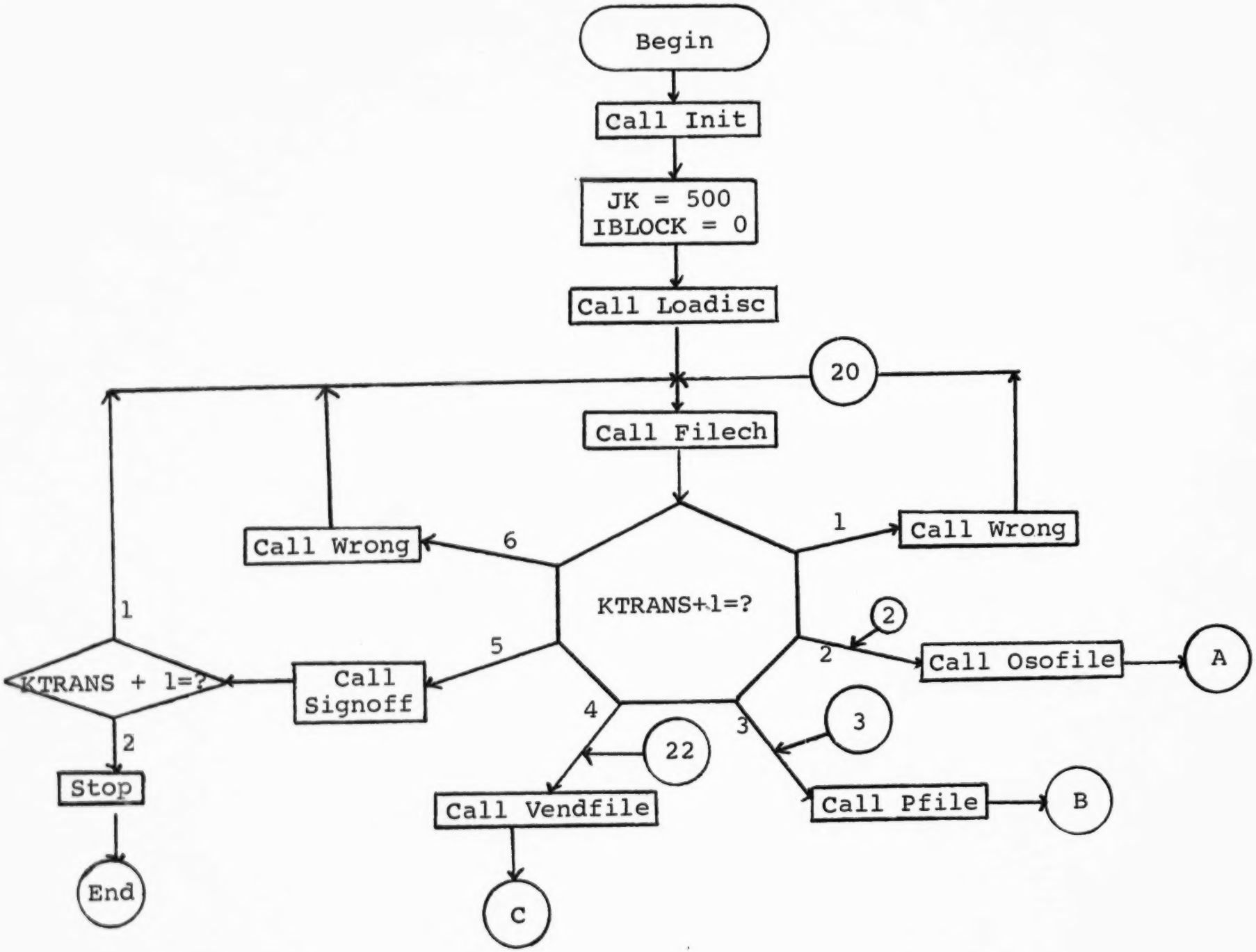
If the user chooses the vendor file the following options are available: 1) correct a record (VCORRECT), 2) delete a record (VDELETE), 3) add a record (VADD), 4) ask for display of a vendor by inputting vendor code (SVCODE), and 5) move to a different file (by returning to FILECH).

If the purchase file is selected the following choices are presented: 1) vendor status - the number of orders currently waiting to be batched to a particular vendor (SVEND),

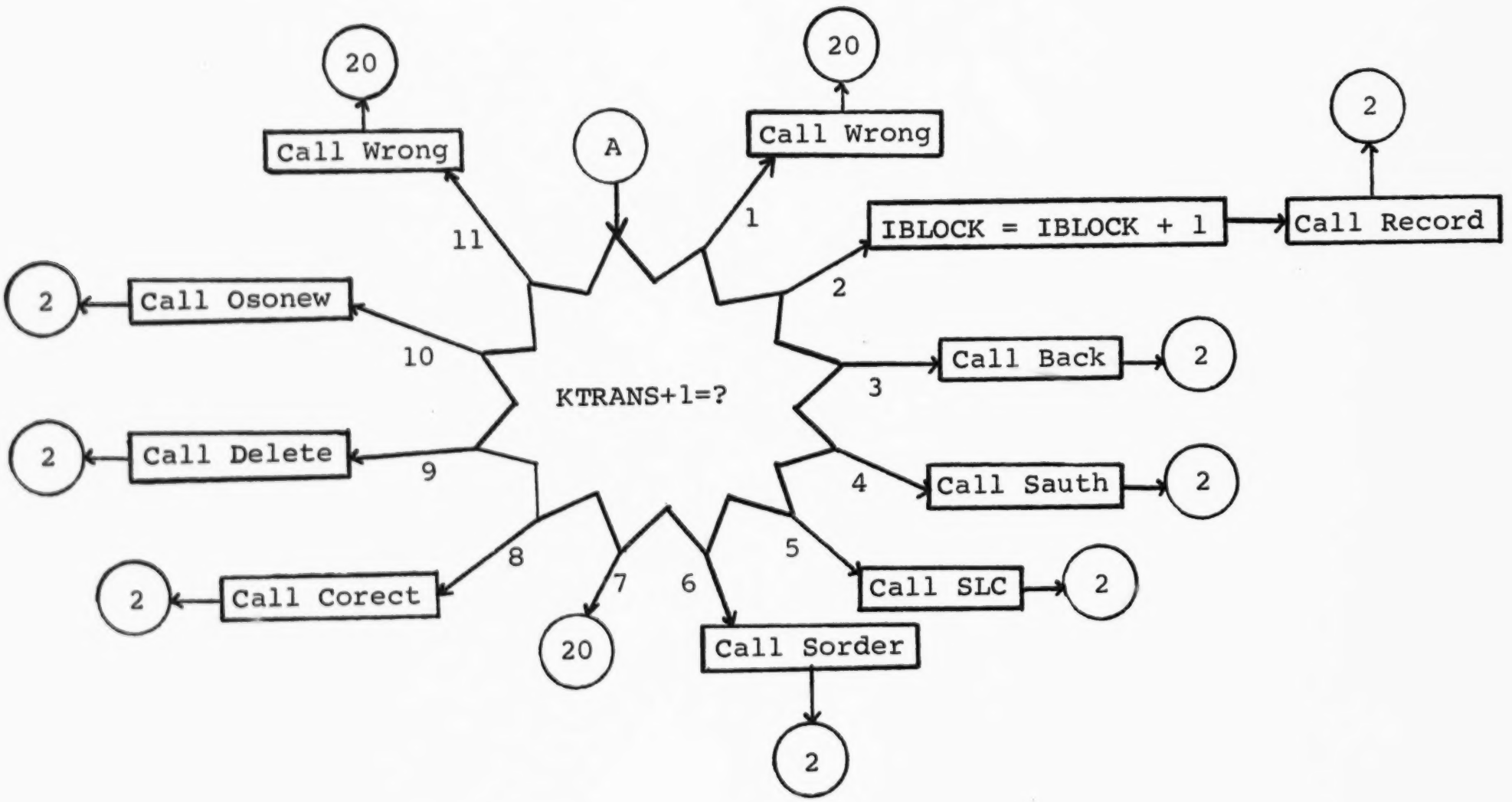
2) hardcopy of the purchase order (PURCHAS), 3) add a new record (NEWREC), 4) search the purchase file by order number (PSORDER), 5) correct an existing record (PURCRDT), 6) delete a record (PURDELT), 7) move to a different file (by returning to FILECH), and 8) search the file by main entry (PSAUTH).

The SIGNOFF action results in a break in on-line contact with the computer.

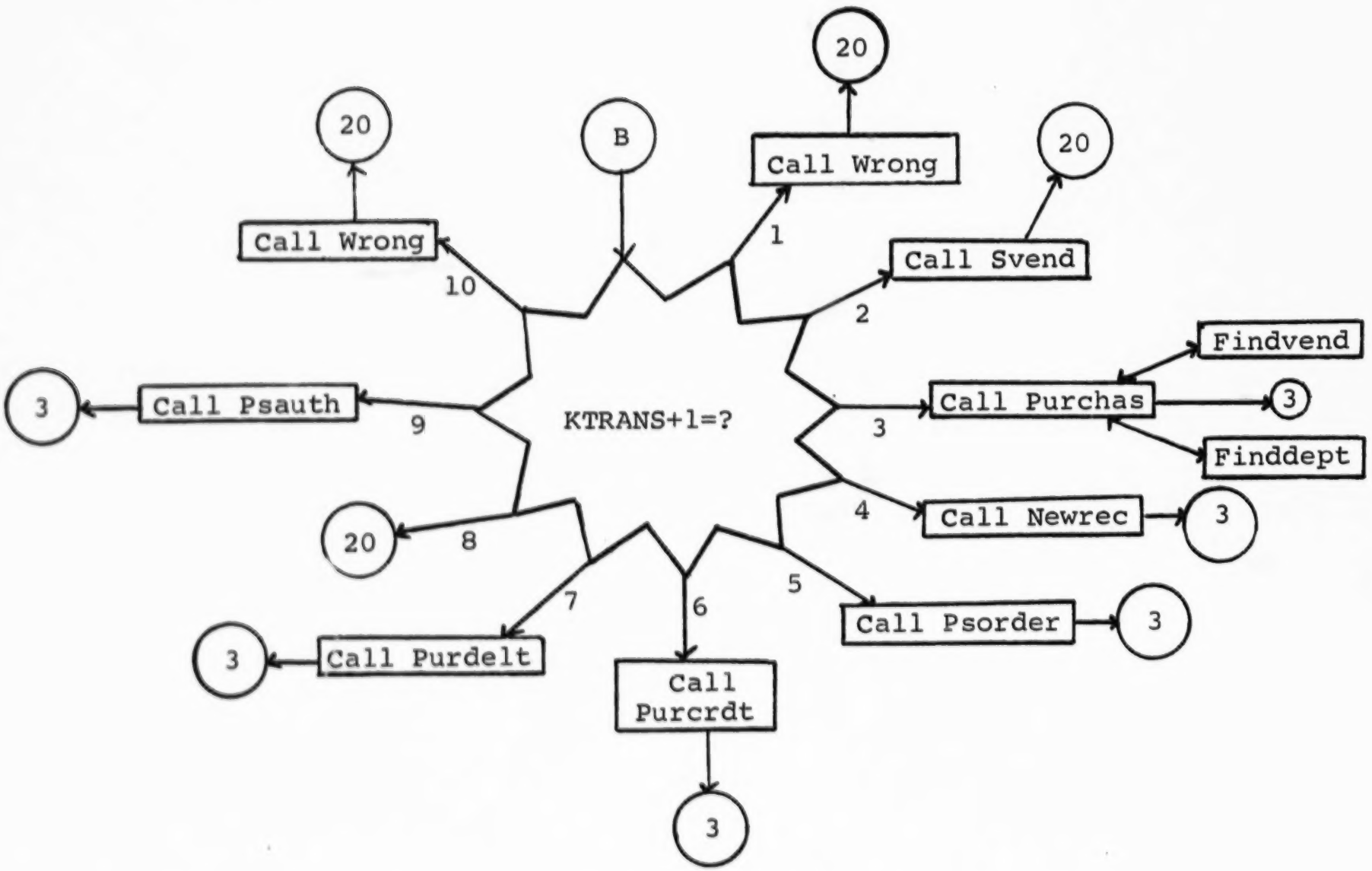
GENERAL PROGRAM



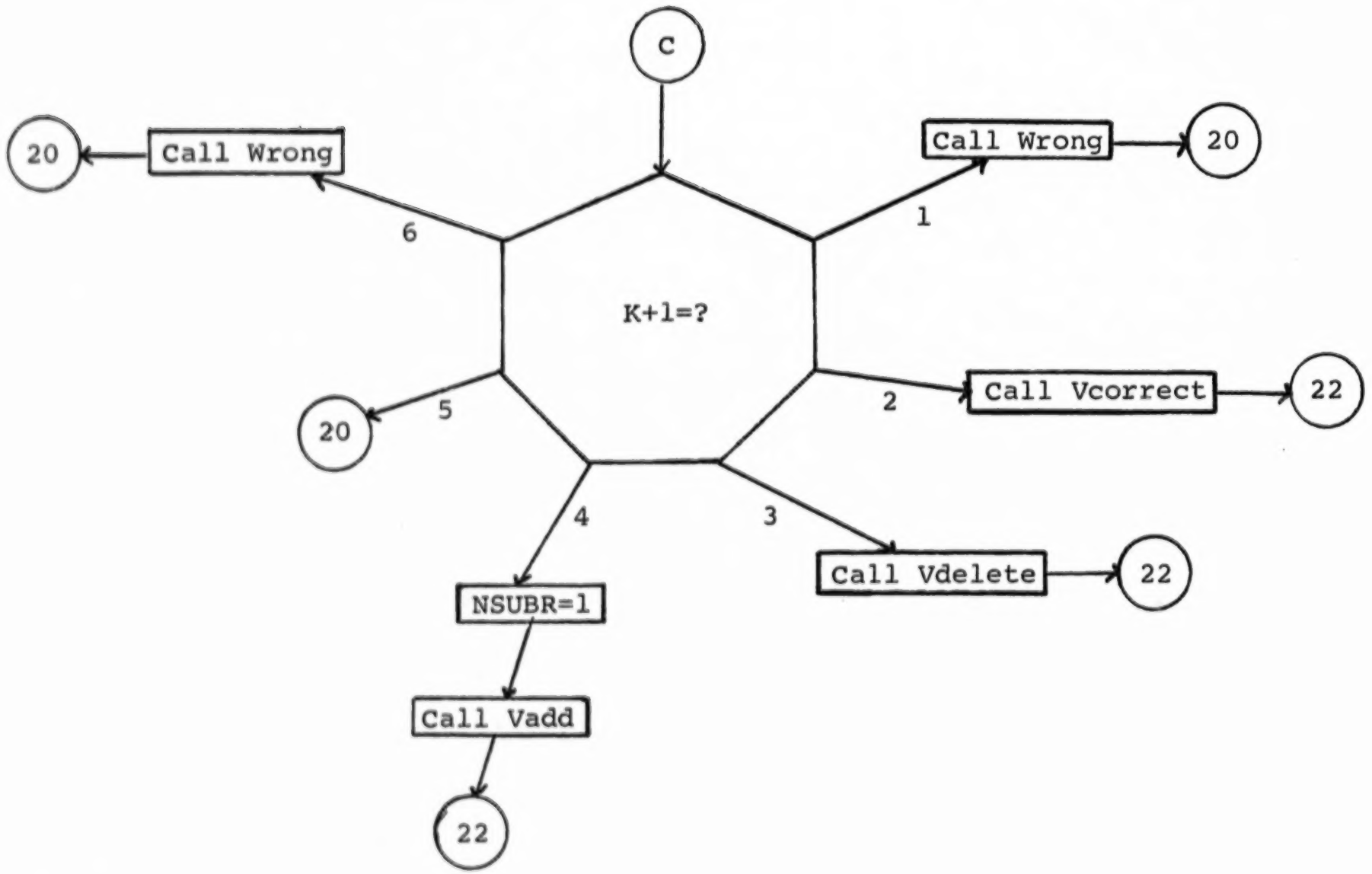
A - Continuation

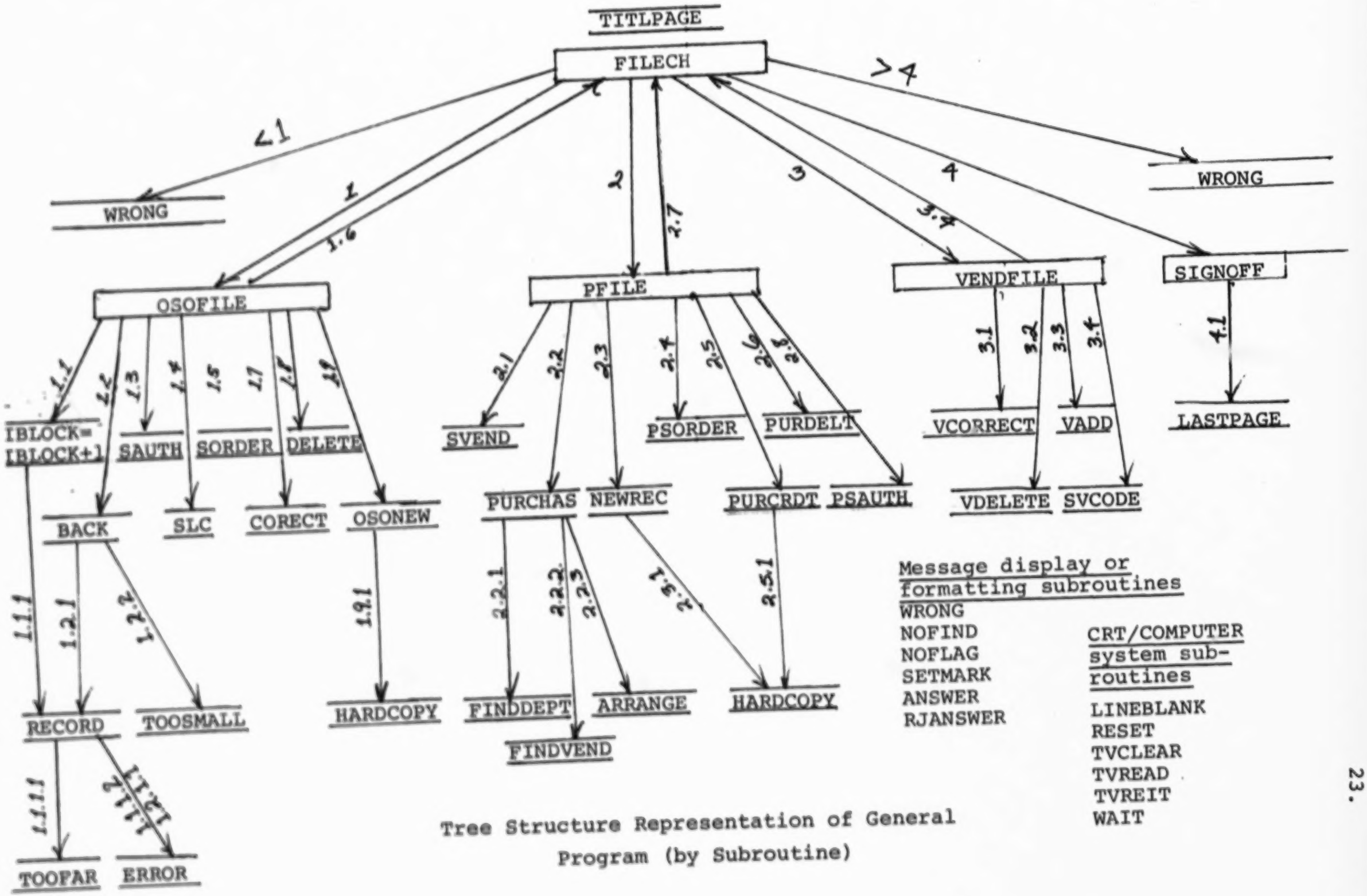


B - Continuation



C - Continuation





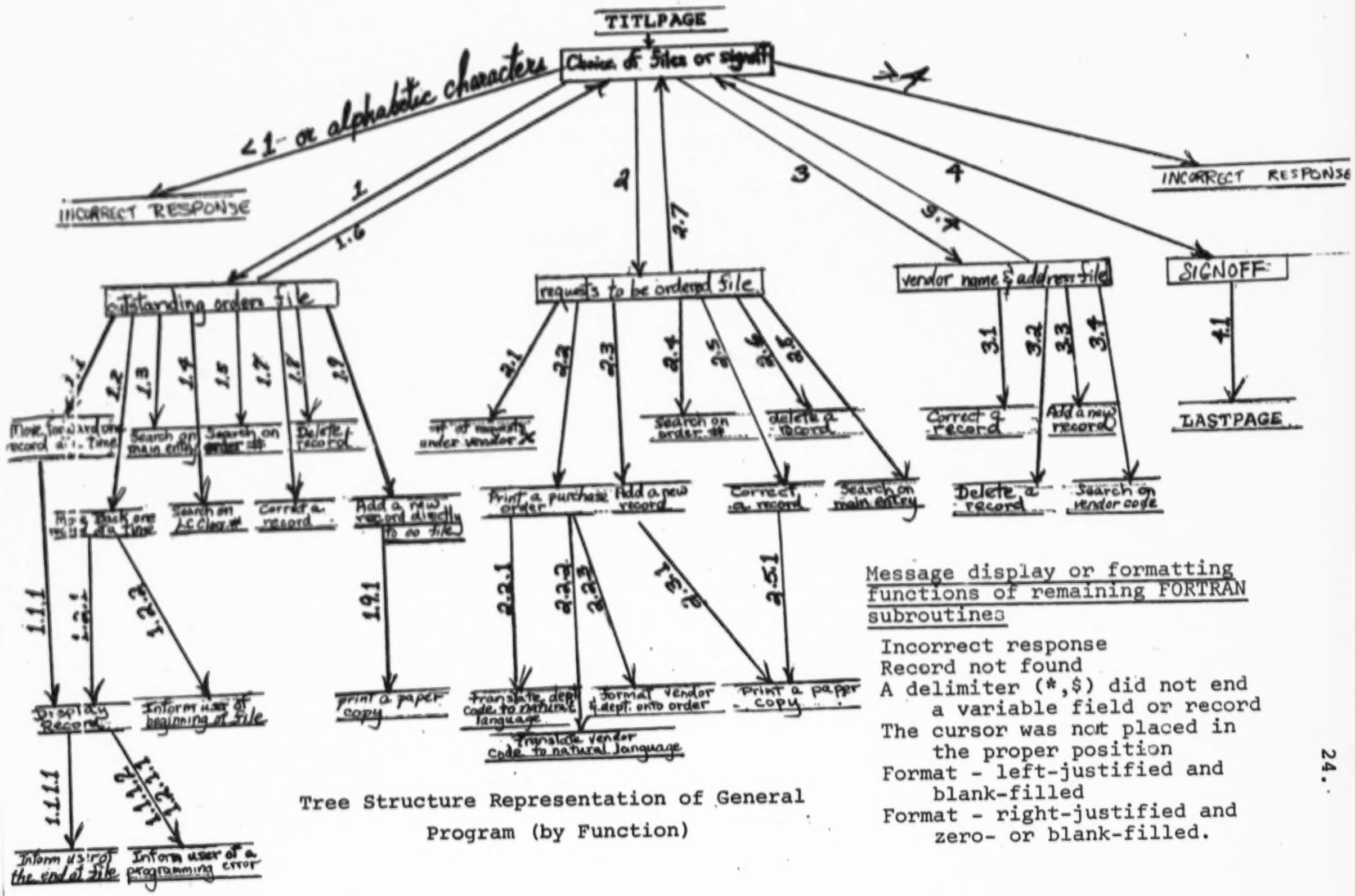
Message display or formatting subroutines

WRONG
NOFIND
NOFLAG
SETMARK
ANSWER
RJANSWER

CRT/COMPUTER system subroutines

LINEBLANK
RESET
TVCLEAR
TVREAD
TVREIT
WAIT

Tree Structure Representation of General Program (by Subroutine)



Tree Structure Representation of General Program (by Function)

Message display or formatting functions of remaining FORTRAN subroutines

Incorrect response
Record not found
A delimiter (*,$) did not end a variable field or record
The cursor was not placed in the proper position
Format - left-justified and blank-filled
Format - right-justified and zero- or blank-filled.

APPENDIX B

SUBROUTINES AVAILABLE
(Alphabetical Order)

Name	Number
ARRANGE	3300
BACK	1000
CORRECT	200
DELETE	1100
ERROR	2800
FILECH(M)	2000
FINDDEPT	3500
FINDVEND	3600
HARDCOPY	3000
INARRAY(IFIR,ISEC)	4400
LASTPAGE	3100
LOADISC	1400
NEMREC	100
NOFIND	700
NOFLAG	2900
OSOFILE(K)	600
OSONEM	1500
PFILE(K)	2100
PSAUTH	3800
PSORDER	1600
PURCHAS	1900
PURCRDT	1700
PURDELT	1800
RECORD	400
RJANSWER	0000
SAUTH	800
SETMARK	2700
SIGNOFF(L)	1300
SLC	900
SORDER	300
SVCODE	3200
SVEND	2200
TITLPAGE	2600
TOOFAR	2300
TOOSMALL	2400
UNAVAIL	500
VADD	4300
VCORRECT	4100
VDELETE	4200
VENDFILE	4000
WRONG	1200

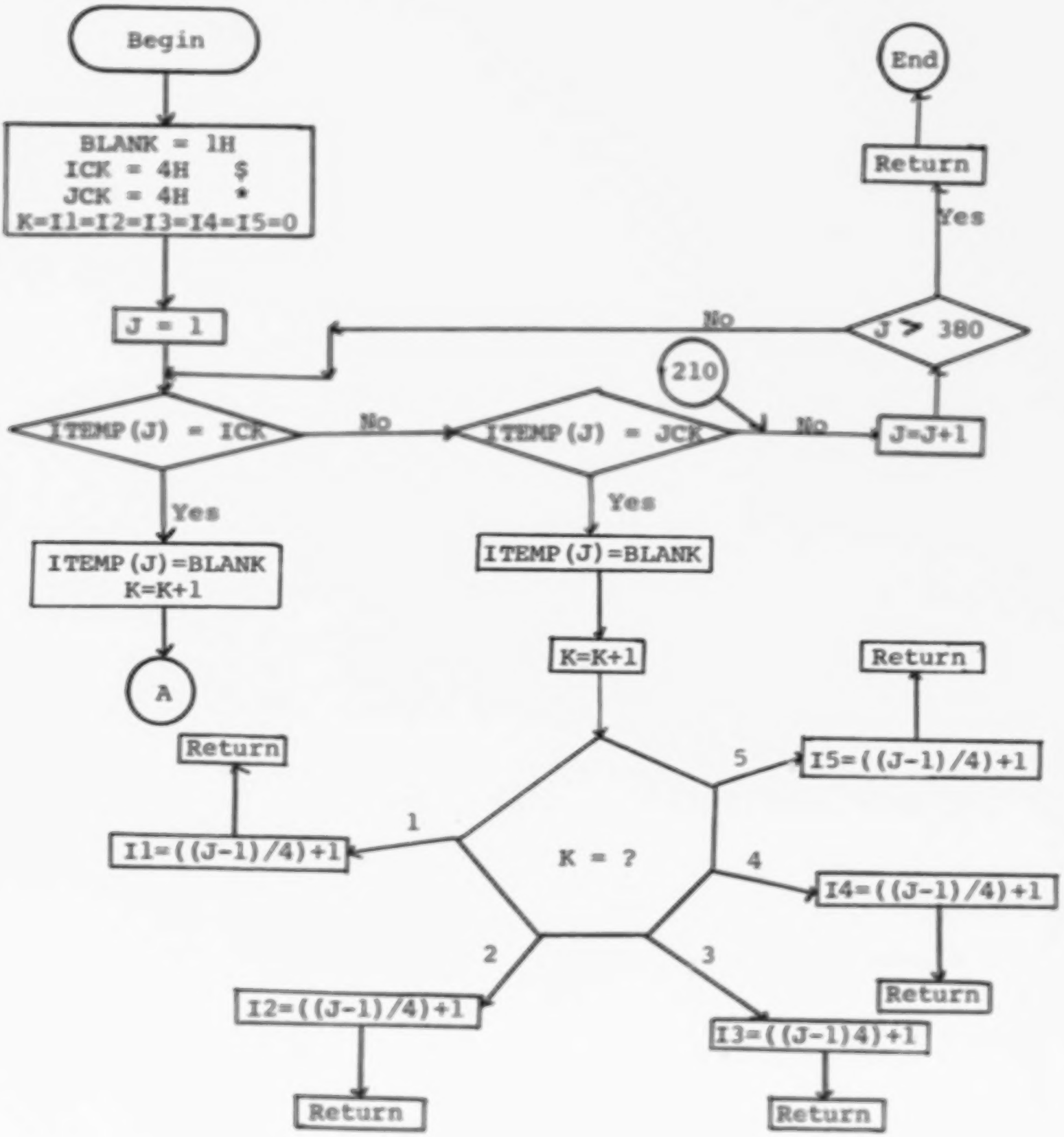
SUBROUTINES AVAILABLE
(Numerical Order)

Name	Number
RJANSWER	000
MEMREC	100
CORECT	200
SORDER	300
RECORD	400
UNAVAIL	500
OSOFILE(K)	600
NOFIND	700
SAUTH	800
SLC	900
BACK	1000
DELETE	1100
WRONG	1200
SIGNOFF(L)	1300
LOADISC	1400
OSONEM	1500
PSORDER	1600
PURCRDT	1700
PURDELT	1800
PURCHAS	1900
FILECH(M)	2000
PFILE(K)	2100
SVEND	2200
TOOFAR	2300
TOOSMALL	2400
TITLPAGE	2600
SETMARK	2700
ERROR	2800
NOFLAG	2900
HARDCOPY	3000
LASTPAGE	3100
SVCODE	3200
ARRANGE	3300
FINDDEPT	3500
FINDVEND	3600
PSAUTH	3800
VENDFILE	4000
VCORRECT	4100
VDELETE	4200
VADD	4300
INARRAY(IFIR,ISEC)	4400

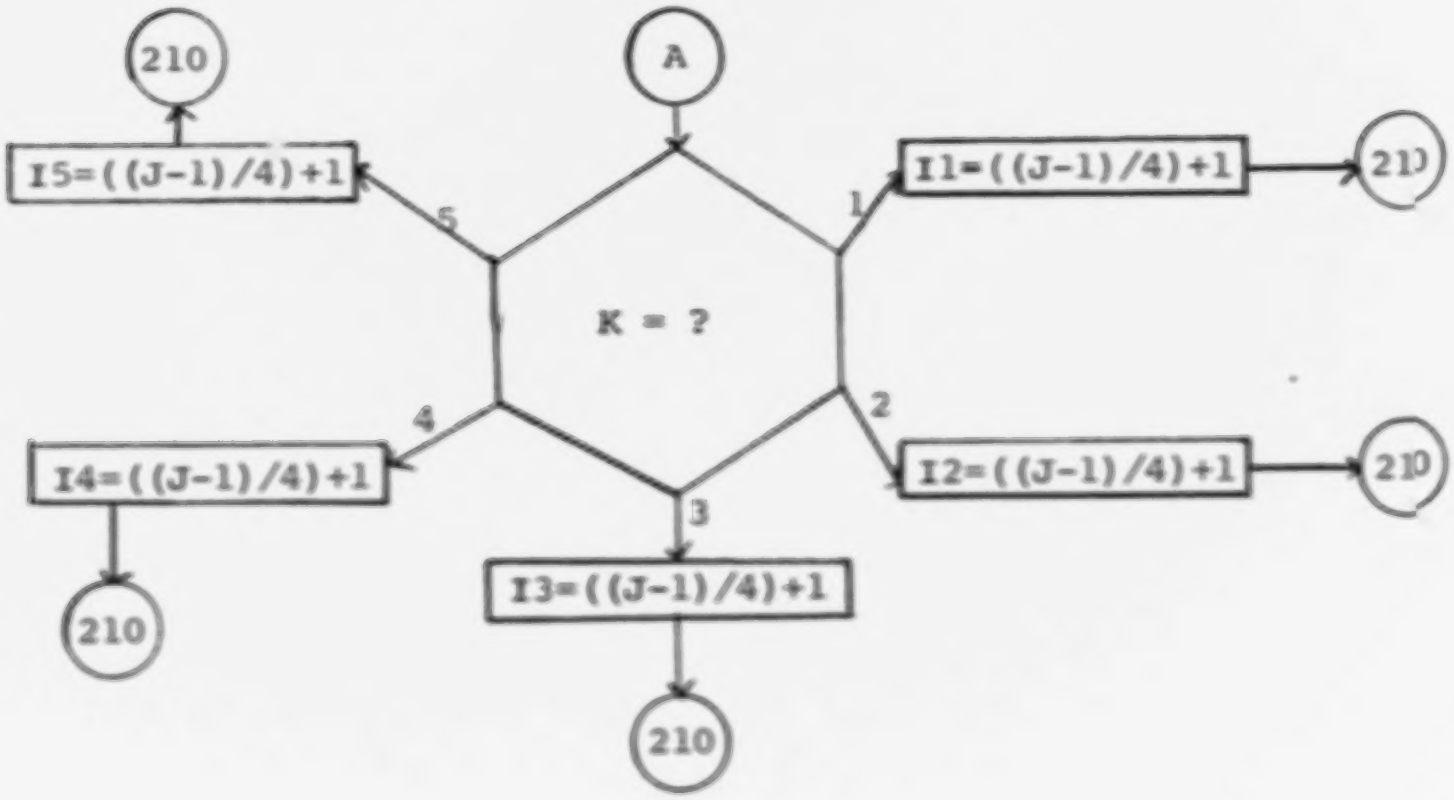
APPENDIX F

SUBROUTINE ARRANGE (I1,I2,I3,I4,I5)

For a variable length of characters, the computer checks each character to see if it is either a $ or an *. If it is a $, the number of this character location is computed and stored, this continues for each $ until an * is reached. When an * is found, the location of it is computed and stored, and all of the locations are transferred to the calling program.



SUBROUTINE ARRANGE(I1,I2,I3,I4,I5) (cont.)

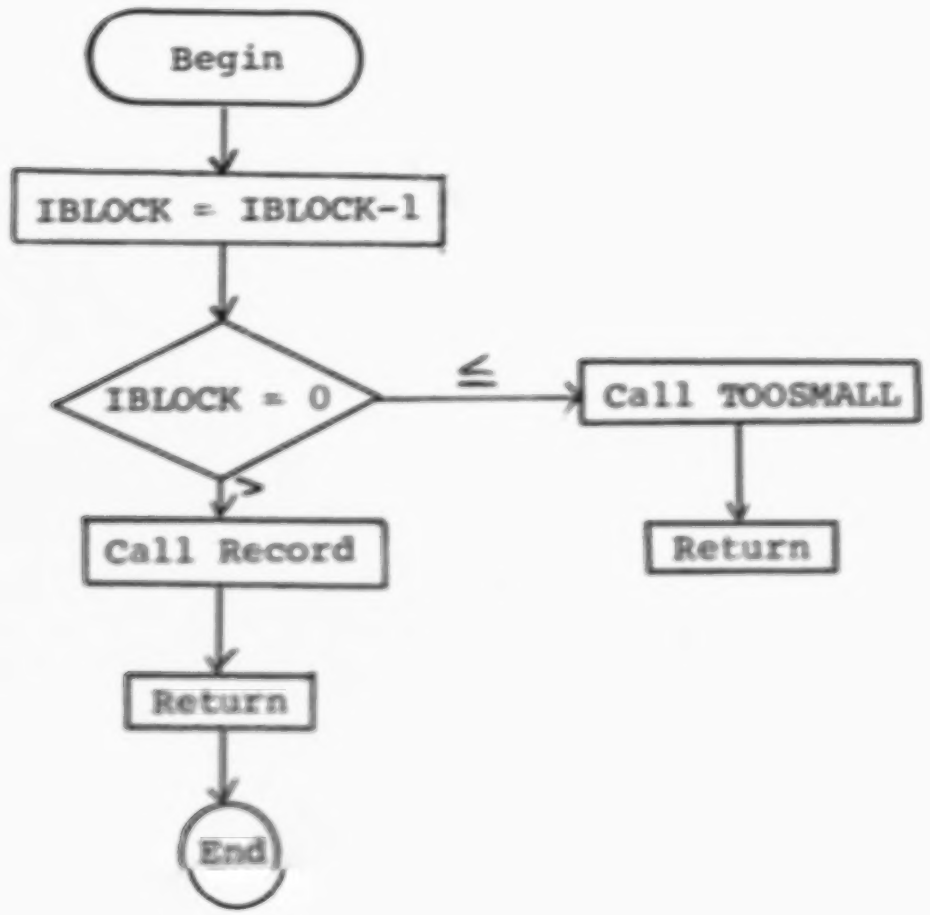


Variables

BLANK = one blank
ICK - location of $ in memory (used as a testing device)
JCK - location of * in memory (used as a testing device)
I1,I2,I3,I4,I5- parameters used to transfer flag locations to the calling program
ITEMP(J)- equivalent to each character in the vendor record
K- counter

SUBROUTINE BACK

Steps backward to next record and checks to see if the beginning of the file has been reached. If not, it displays the record; if it has gone beyond the first record, it calls TOOSMALL.

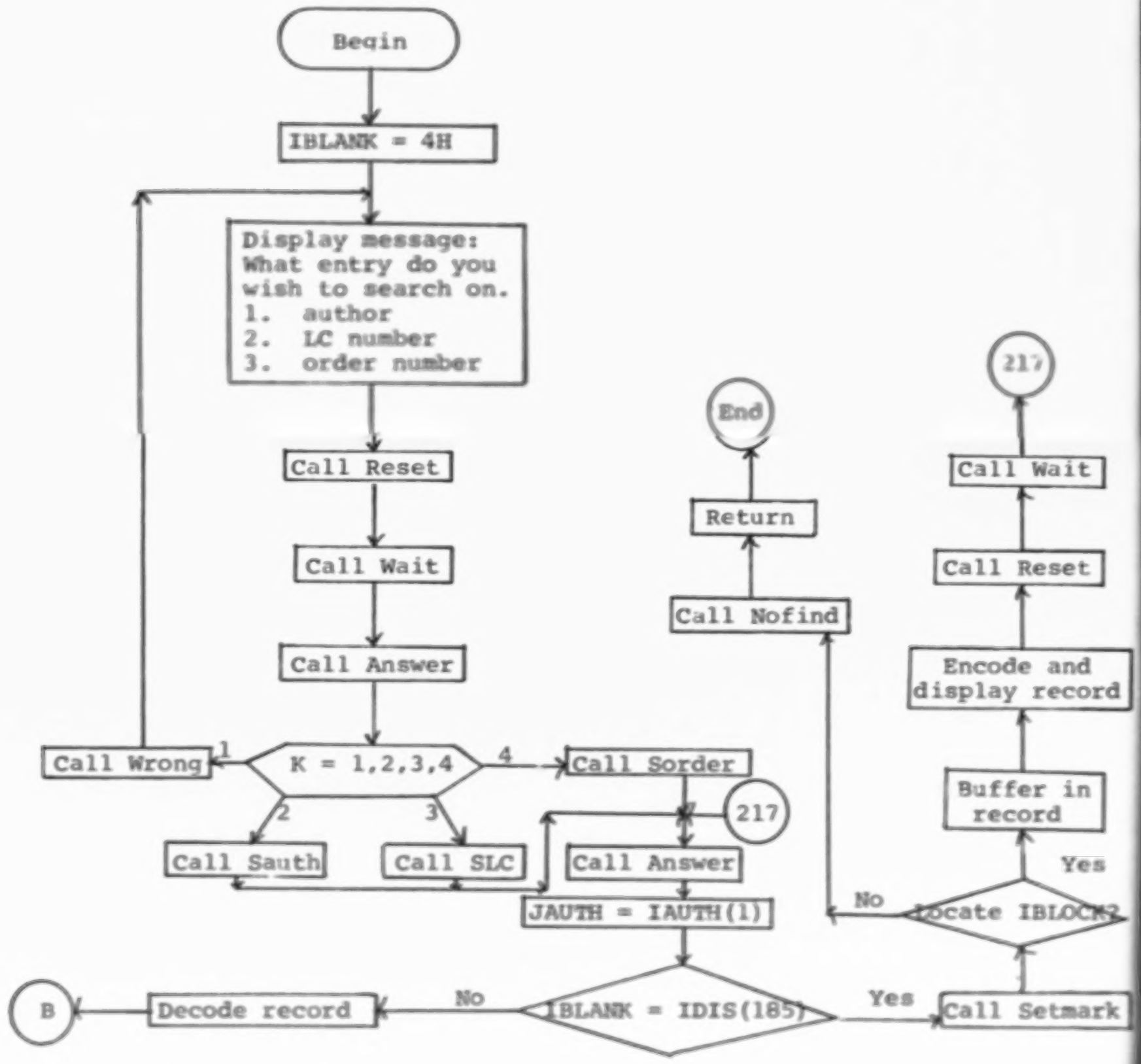


Variables

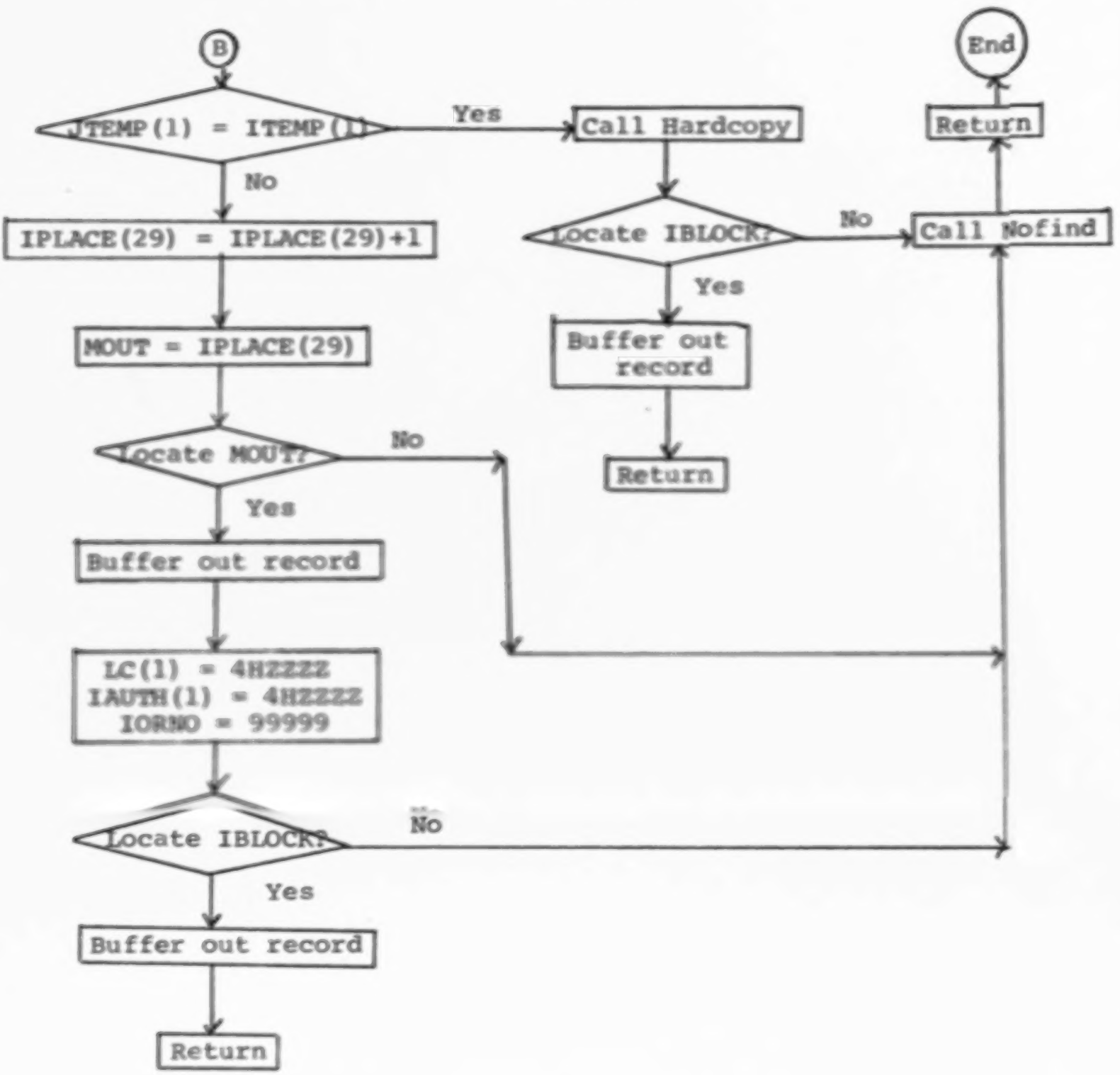
IBLOCK - block location of desired record.

SUBROUTINE CORECT

User chooses which entry he wishes to search on. Once the record has been found, the computer checks to see if the marker is after the last piece of information. If not, Setmark is called and the record is redisplayed. If it is all right, the computer checks to see if the author's name begins with the same letter as it had been before any changes. If it is the same, a hardcopy is made and the record is buffered onto the disc. If it is not the same, the record is transferred to the end of the file and the record in the old location is deleted.



SUBROUTINE CORECT (cont.)



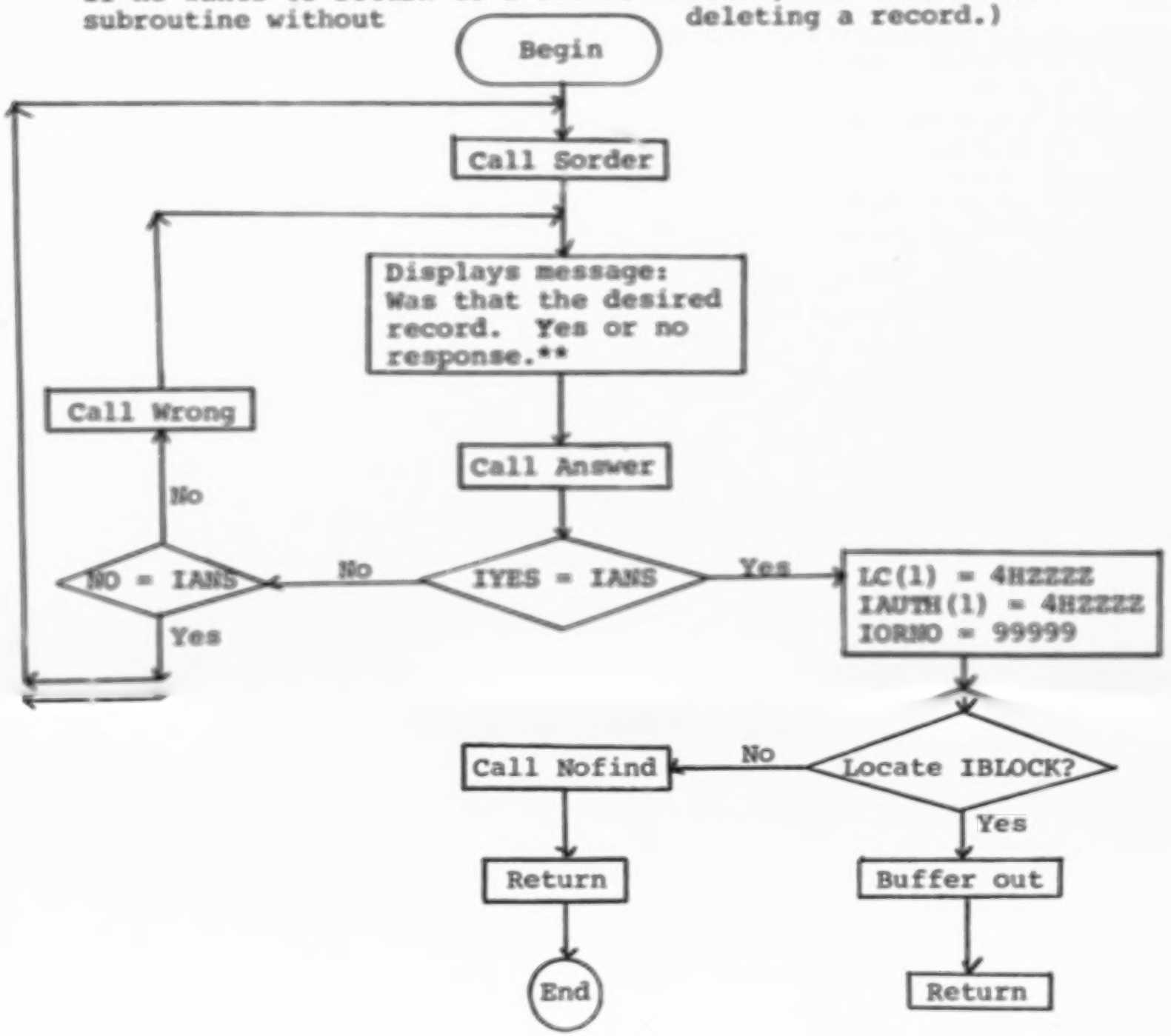
SUBROUTINE CORECT (cont.)

Variables

CHO - memory display buffer
K - storage buffer
JAUTH - equivalent to first letter of author's name on the desired record
IDIS(185) - location of middle of order number in the record on the display screen
IDIS - memory display buffer
JTEMP(1) - equivalent to first letter of author's name
ITEMP(1) - equivalent to first letter of author's name
IPLACE(29) = MOUT - location of last record of the file
IBLOCK - location of desired record

SUBROUTINE DELETE

Asks if a displayed record is the desired record. If it is, it is deleted; if not, the process of searching and displaying is repeated.**(Note: the user can now type in MOVE if he wants to return to a set of choices, and leave this subroutine without deleting a record.)

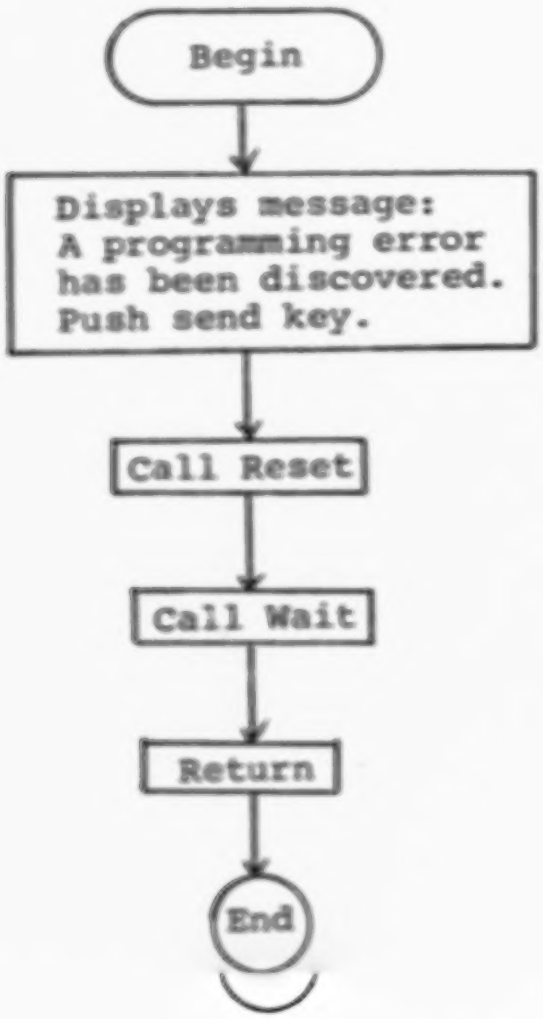


Variables

IRES - memory display buffer
IBLOCK - location of desired record
IANS - storage buffer

SUBROUTINE ERROR

Displays a message.

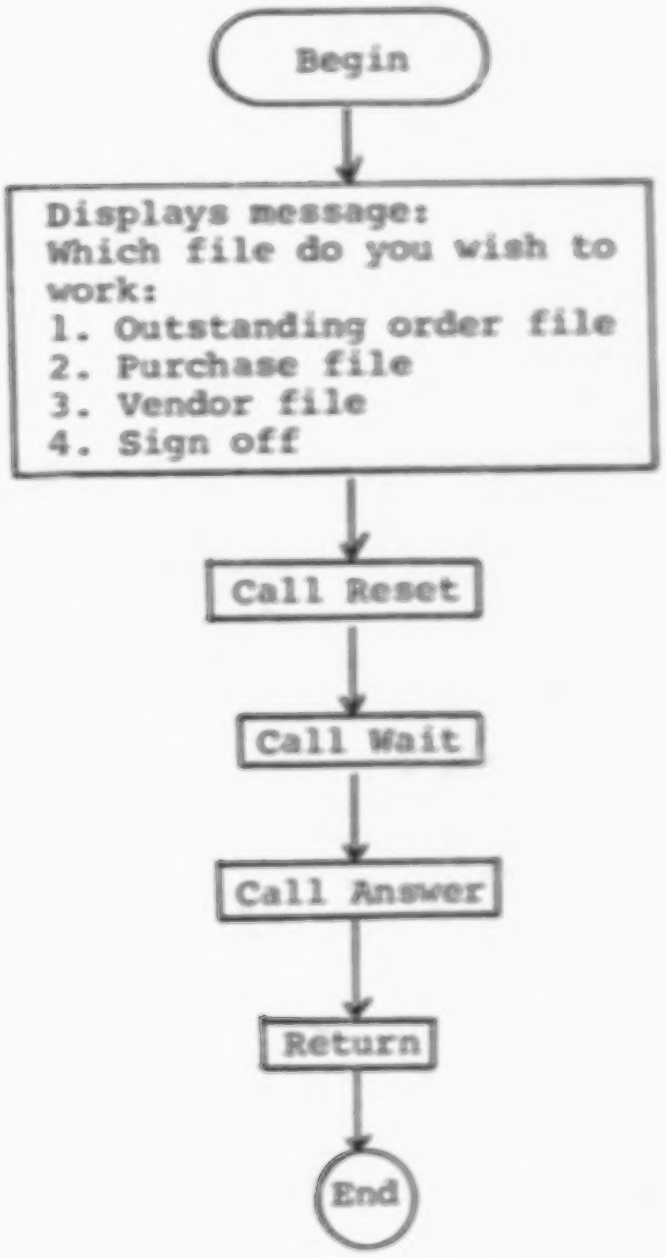


Variables

IREP - memory display buffer

SUBROUTINE FILECH(M)

Displays a message.

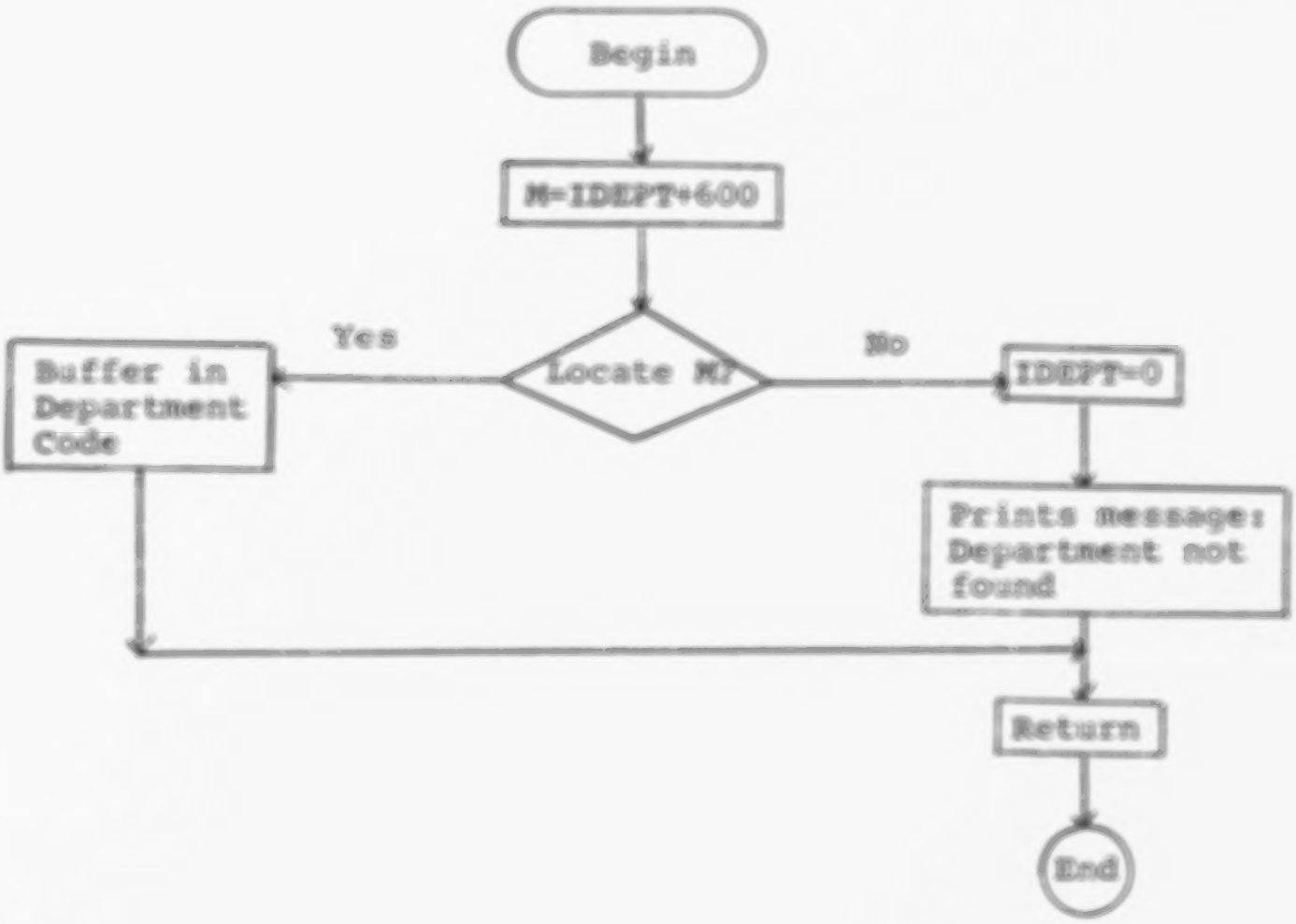


Variables

M - storage buffer
IDIS - memory display buffer

SUBROUTINE FINDDEPT

Searches for a desired department code. If it is found, it is buffered into memory: if it is not found the message: Department Not Found, is displayed.

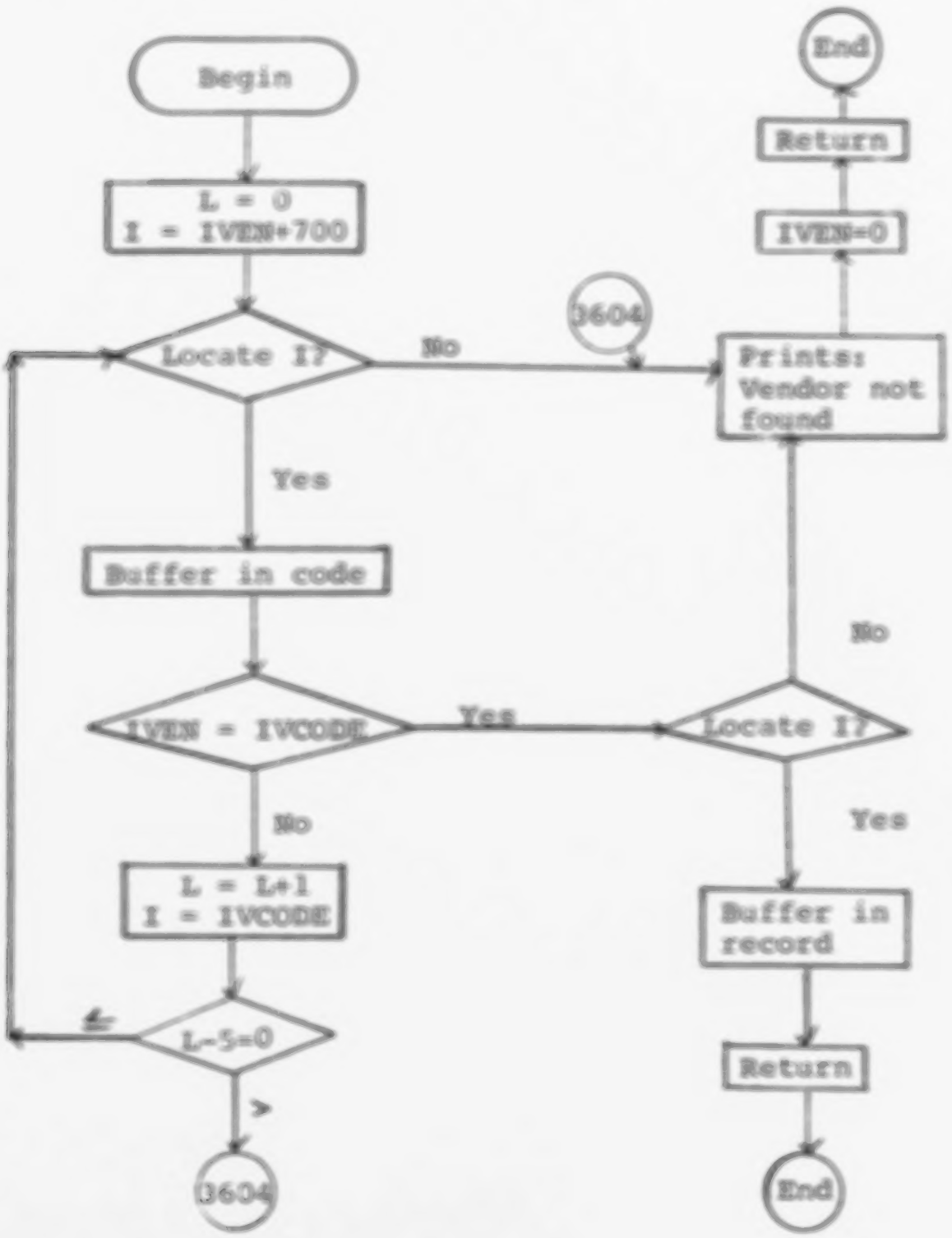


Variables

M - location of desired department code
IDEPT - department code desired by user

SUBROUTINE FINDVEND

Searches for a particular vendor. If it is found, the record is buffered into memory; if it is not found, the message: vendor not found, is printed.

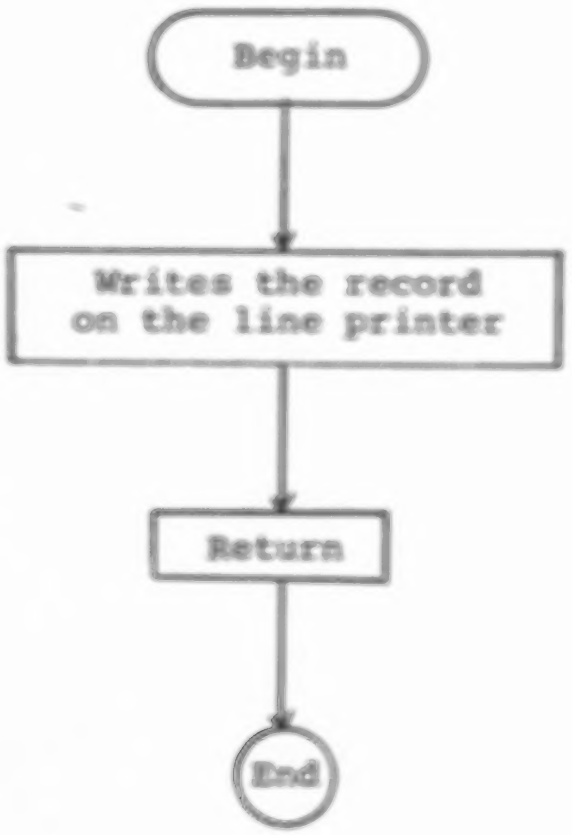


Variables

L - counter
IVEN - vendor code desired by user
IVCODE - vendor code on record

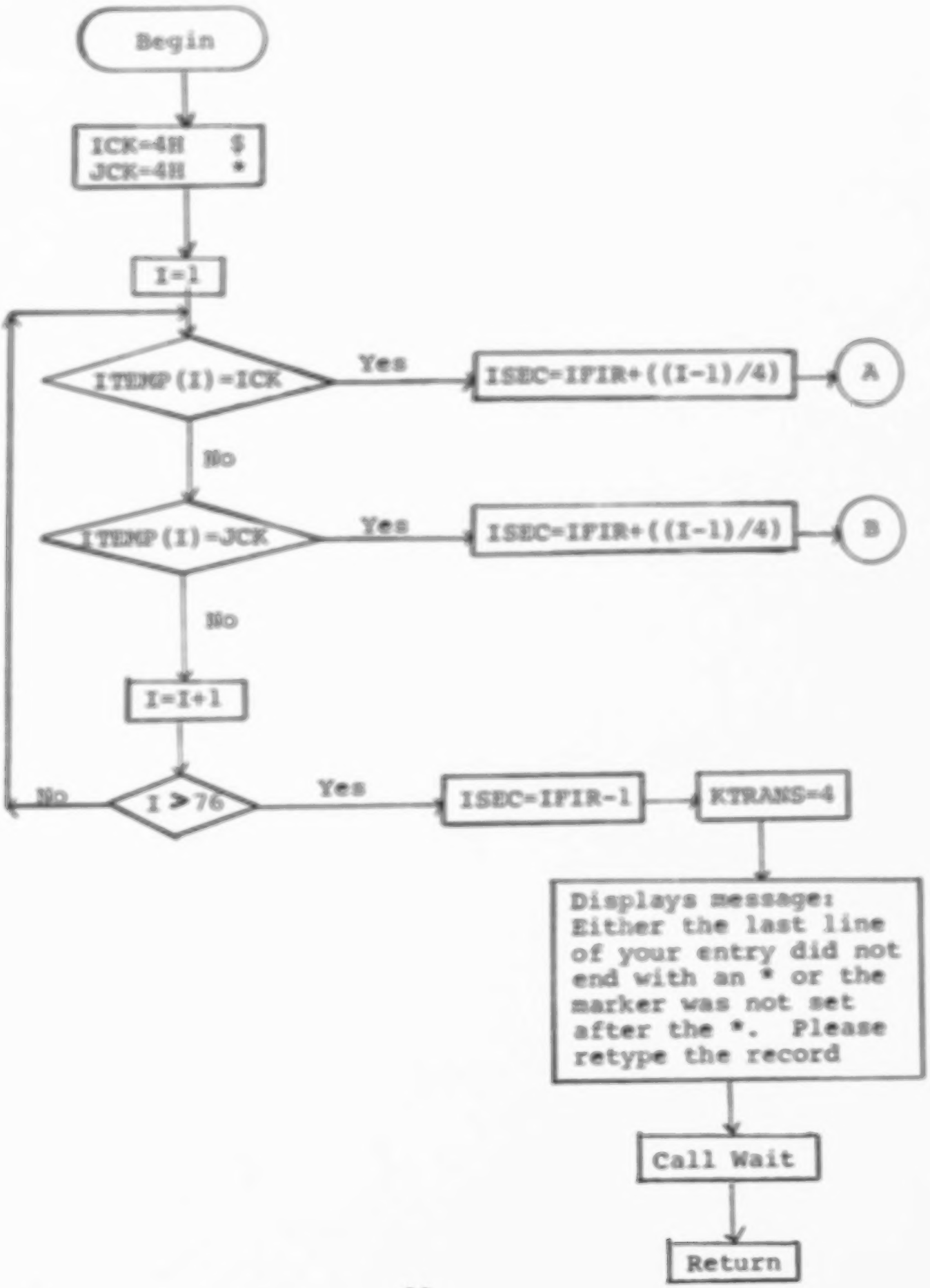
SUBROUTINE HARDCOPY

Types a record on the line printer.

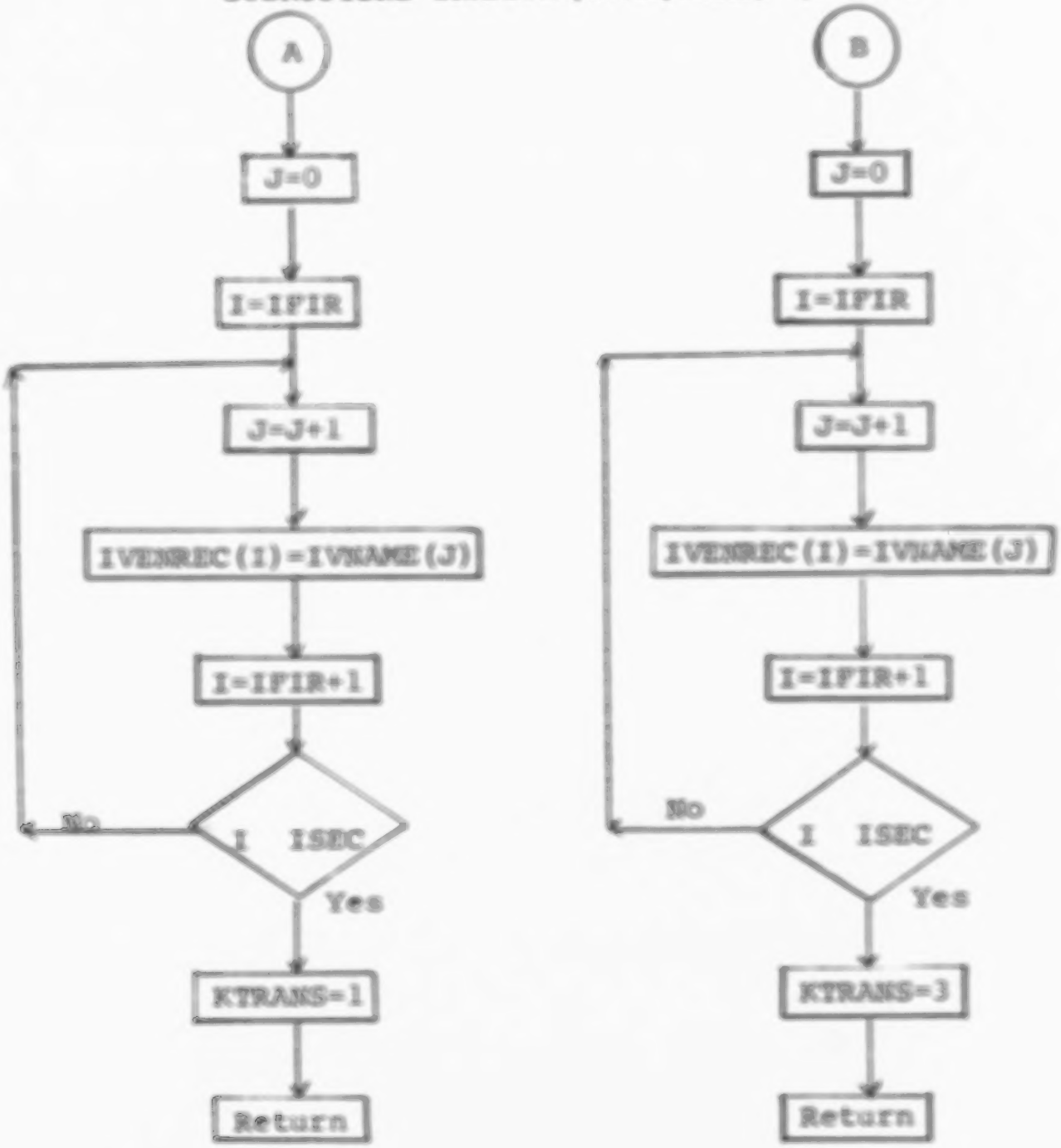


SUBROUTINE INARRAY(IFIR,ISEC)

Rearranges a vendor record in memory so that the disc may be packed as much as possible. Each line of the record is restored into the variable IVENREC(I) thus eliminating undesired blank locations.



SUBROUTINE INARRAY(IFIR,ISEC) (cont.)

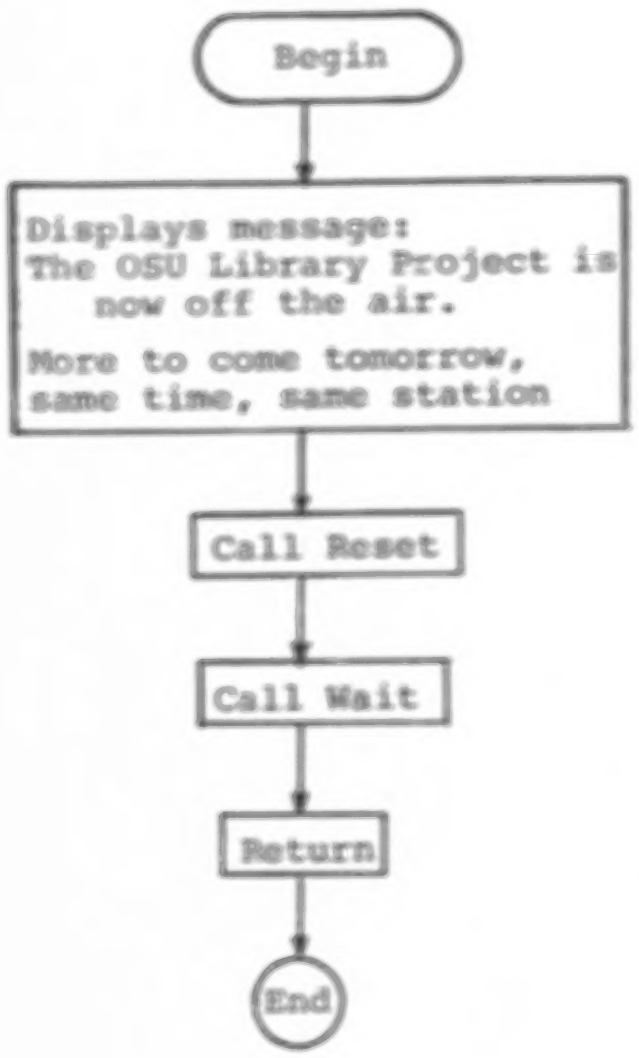


Variables

KTRANS - variable specifying whether or not a flag has been found and whether or not it is an *.
IFIR - location of first character of each line
ISEC - location of flag
ICK - location of $ in memory
JCK - location of * in memory
J - counter
IVENREC(I) - array with entire vendor record
IVNAME(J) - array with a line of the vendor record

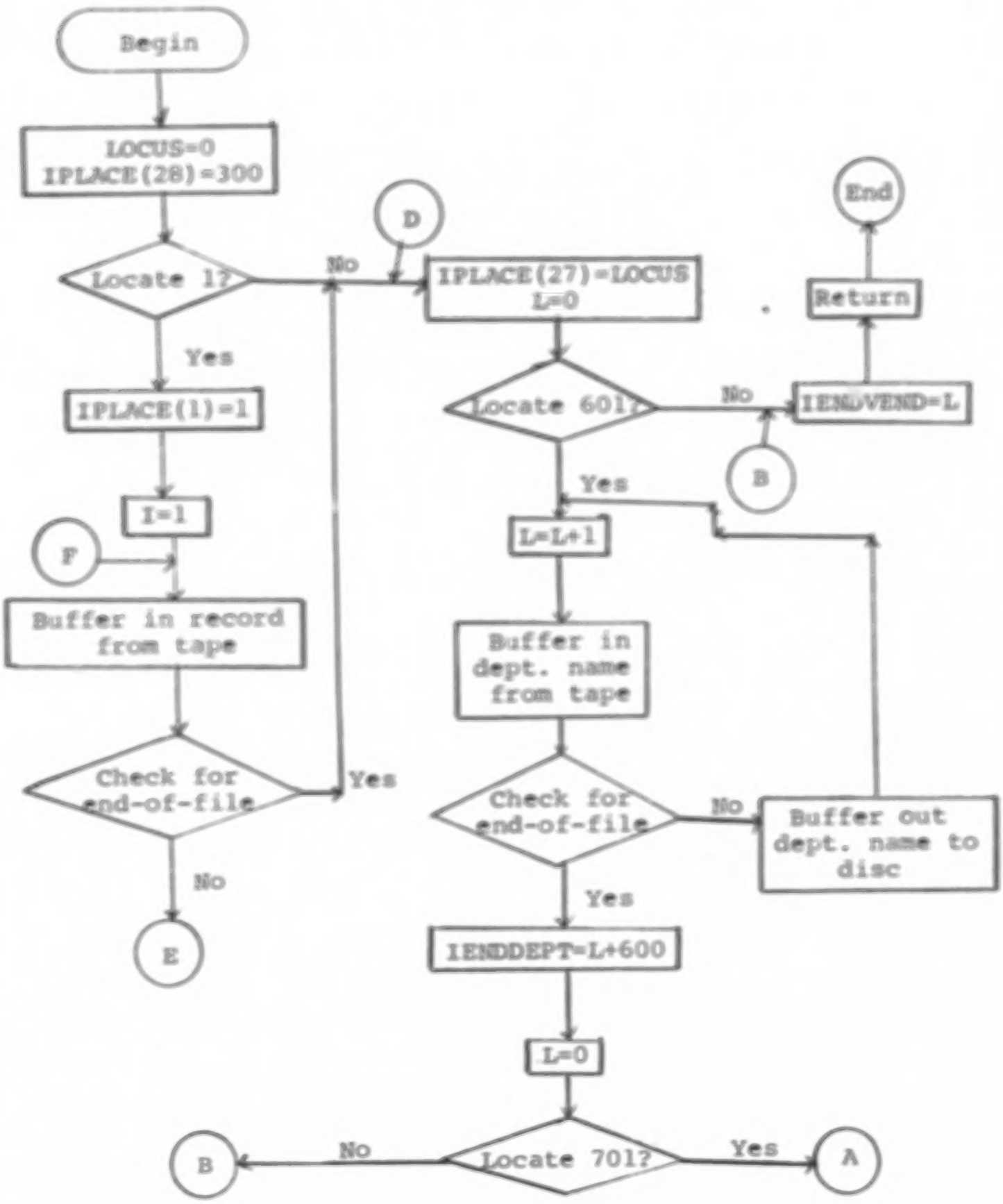
SUBROUTINE LASTPAGE

Displays a message

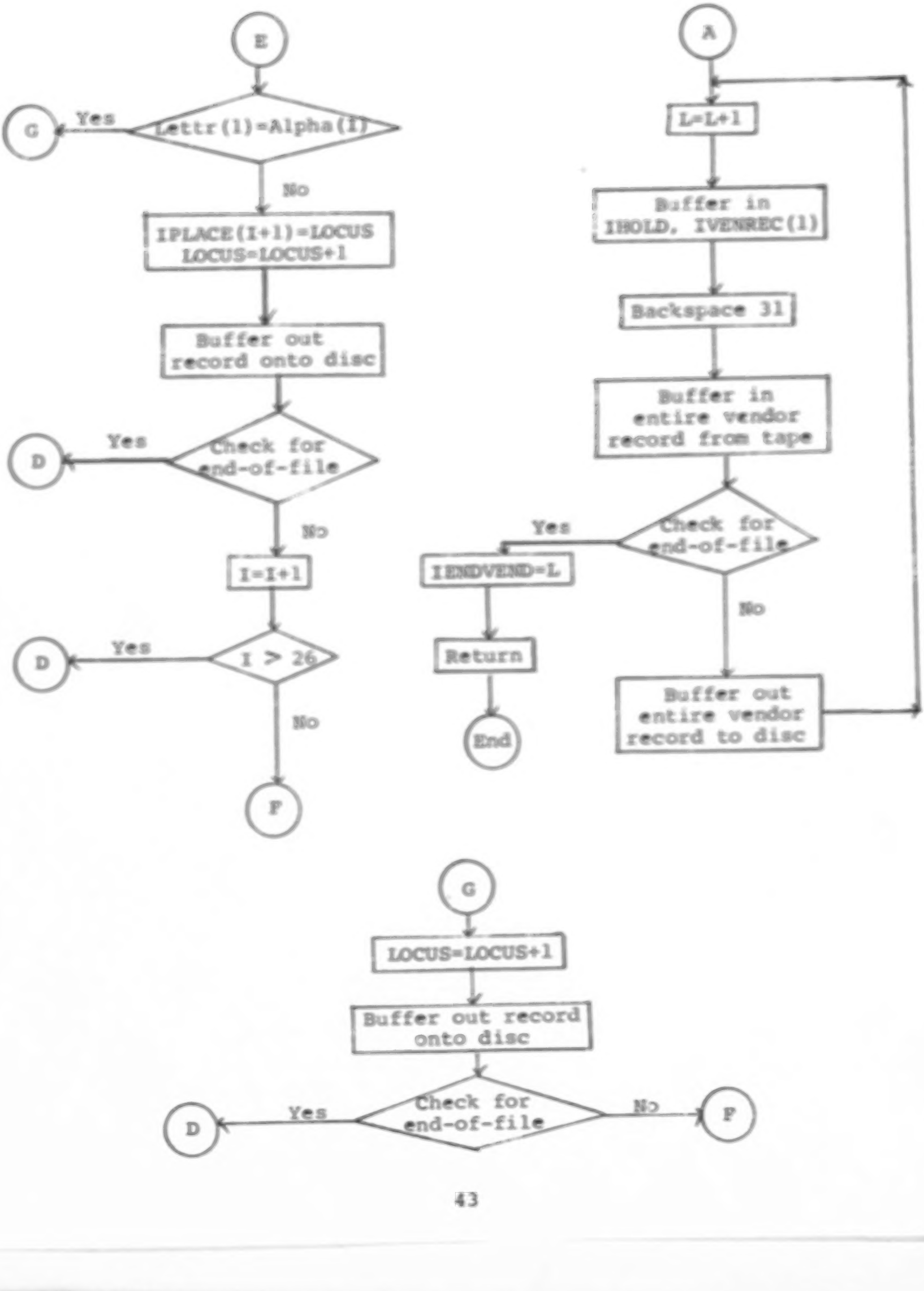


SUBROUTINE LOADISC

Loads the department file, vendor file, and outstanding order file from tape onto disc.



SUBROUTINE LOADISC (cont.)



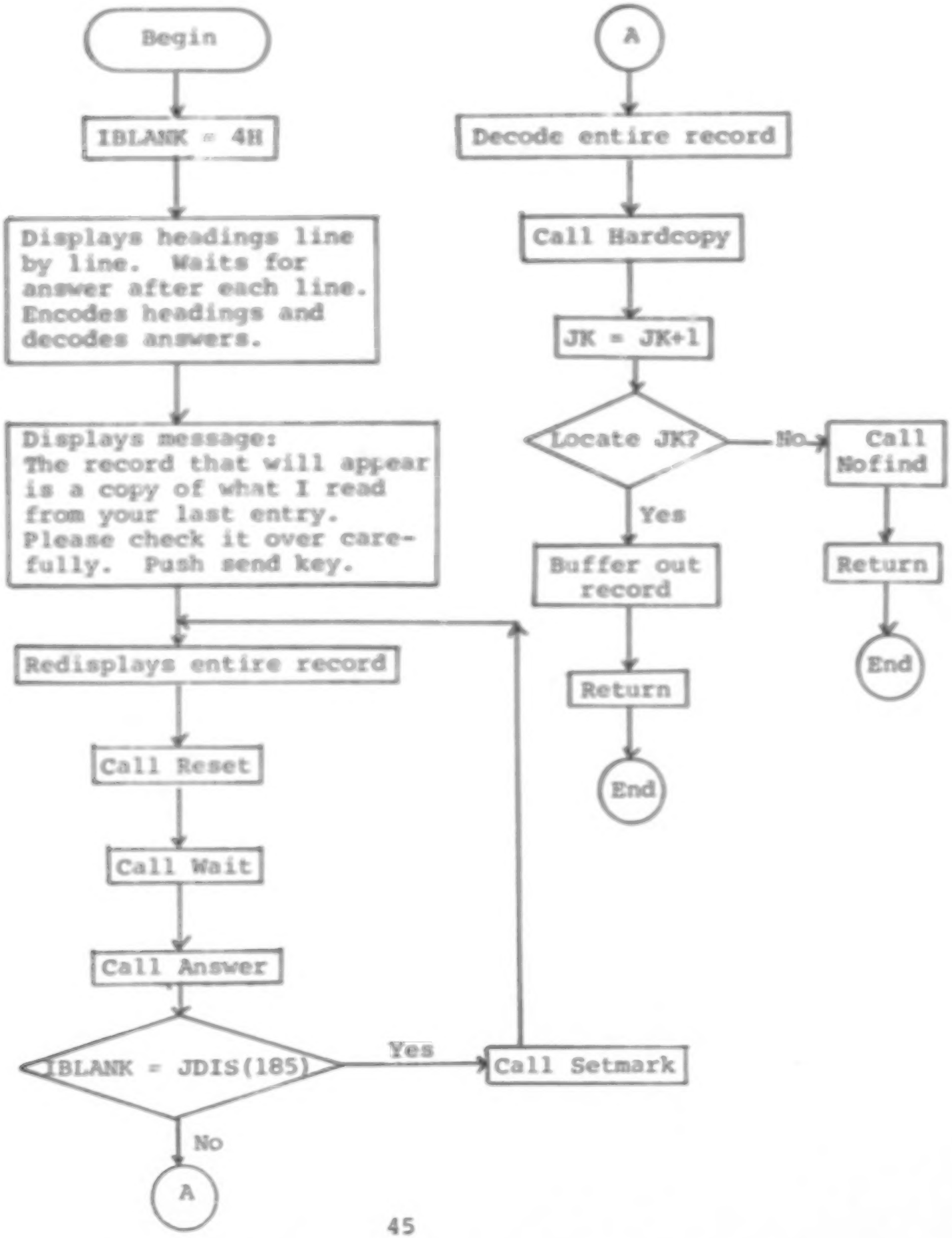
SUBROUTINE LOADISC (cont.)

Variables

LOCUS - counter
IPLACE(28)=300 - location of first record of the outstanding order file new records.
L - counter
601 - first location of department file
1 - first location of outstanding order file
IENDDEPT - location of first available space after department file
701 - first location of vendor file
IHOLD - variable specifying length of variable field for each vendor record
IVENREC - array containing vendor record
IENDVEND - location of first available space after vendor file
IVENREC(IHOLD) - last word of vendor record
LETTR(1) - first letter of main entry
ALPHA(I) - array containing the alphabet
IPLACE(I+1) - location of last record on each alphabetical block
INDEPT(1) - first word of department name

SUBROUTINE NEWREC

Operator types in new record, line by line. The record is redisplayed for any necessary changes. The computer checks to see if the marker is after the last piece of information. If it is, a hardcopy is printed out. If not, the computer re-displays the new record previous to corrections.



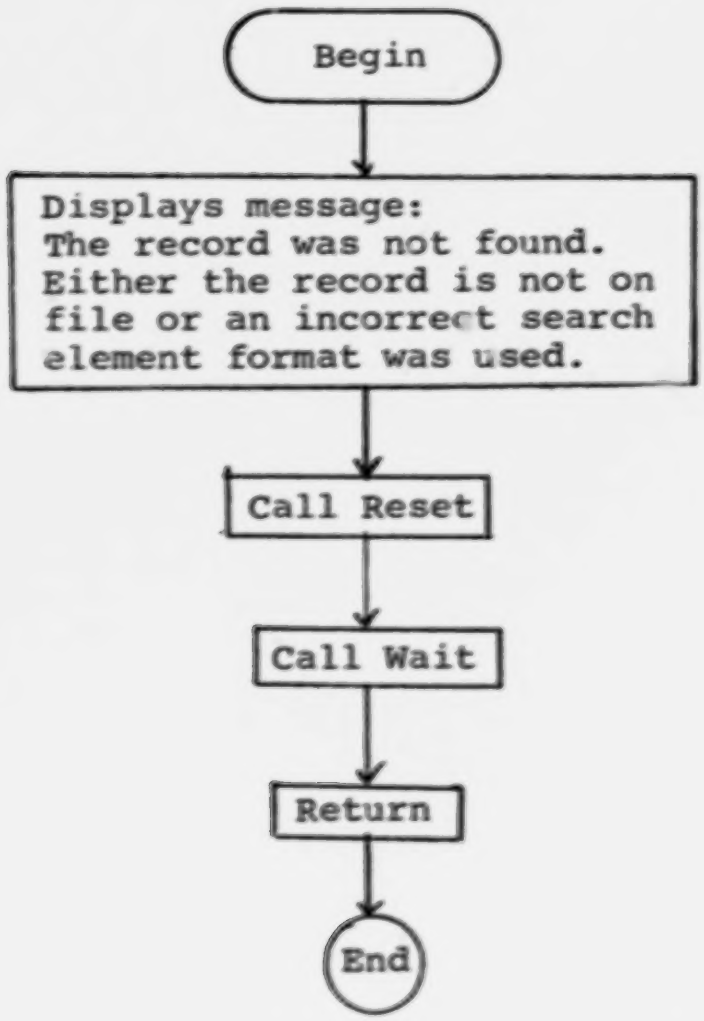
SUBROUTINE NEWREC (cont.)

Variables

JK - location of last record on the file
IDIS - memory display buffer
JDIS - storage buffer
JDIS(185) - location of middle of order number in the record
   being displayed
MESS - memory display buffer

SUBROUTINE NOFIND

Displays a message.

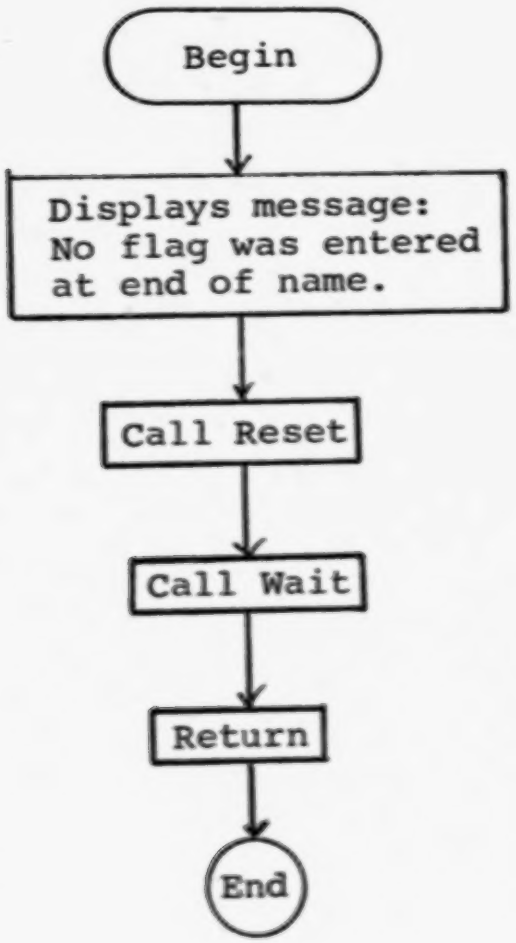


Variables

IERR - memory display buffer

SUBROUTINE NOFLAG

Displays a message.

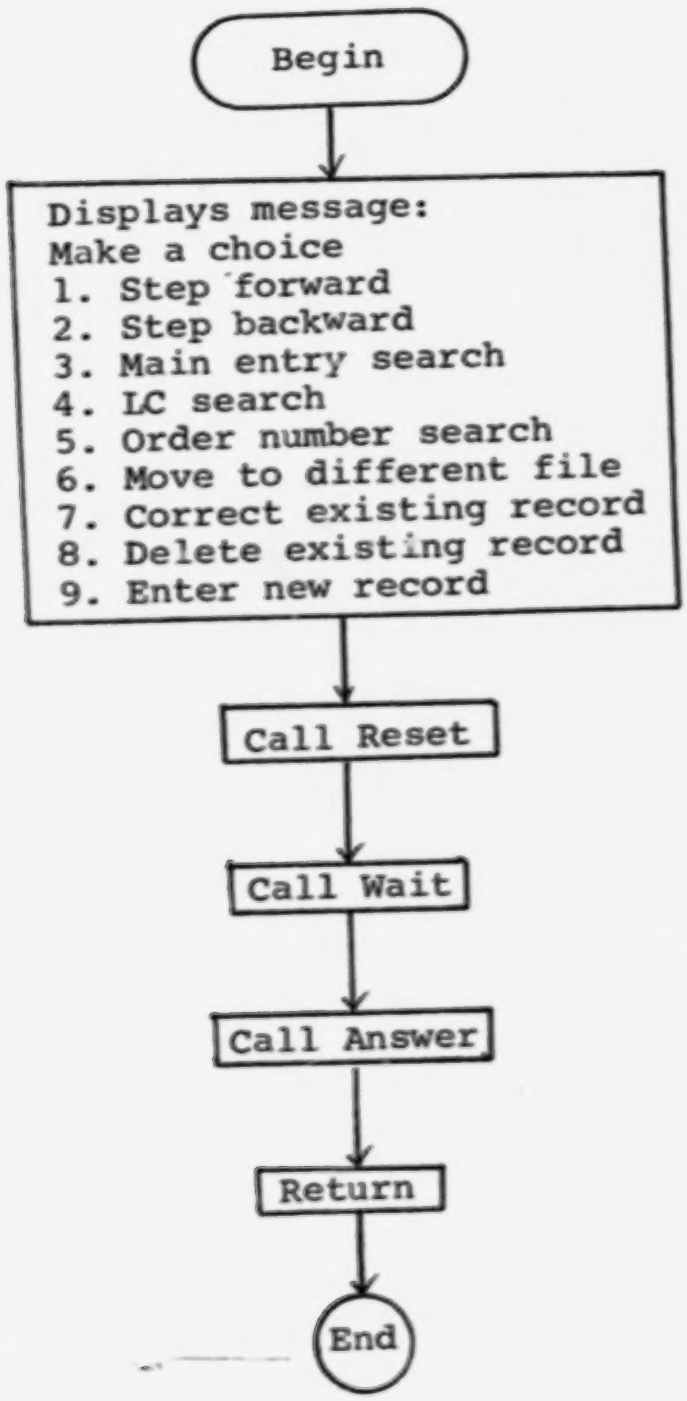


Variables

IRES - memory display buffer

SUBROUTINE OSOFILE(K)

Displays a message.

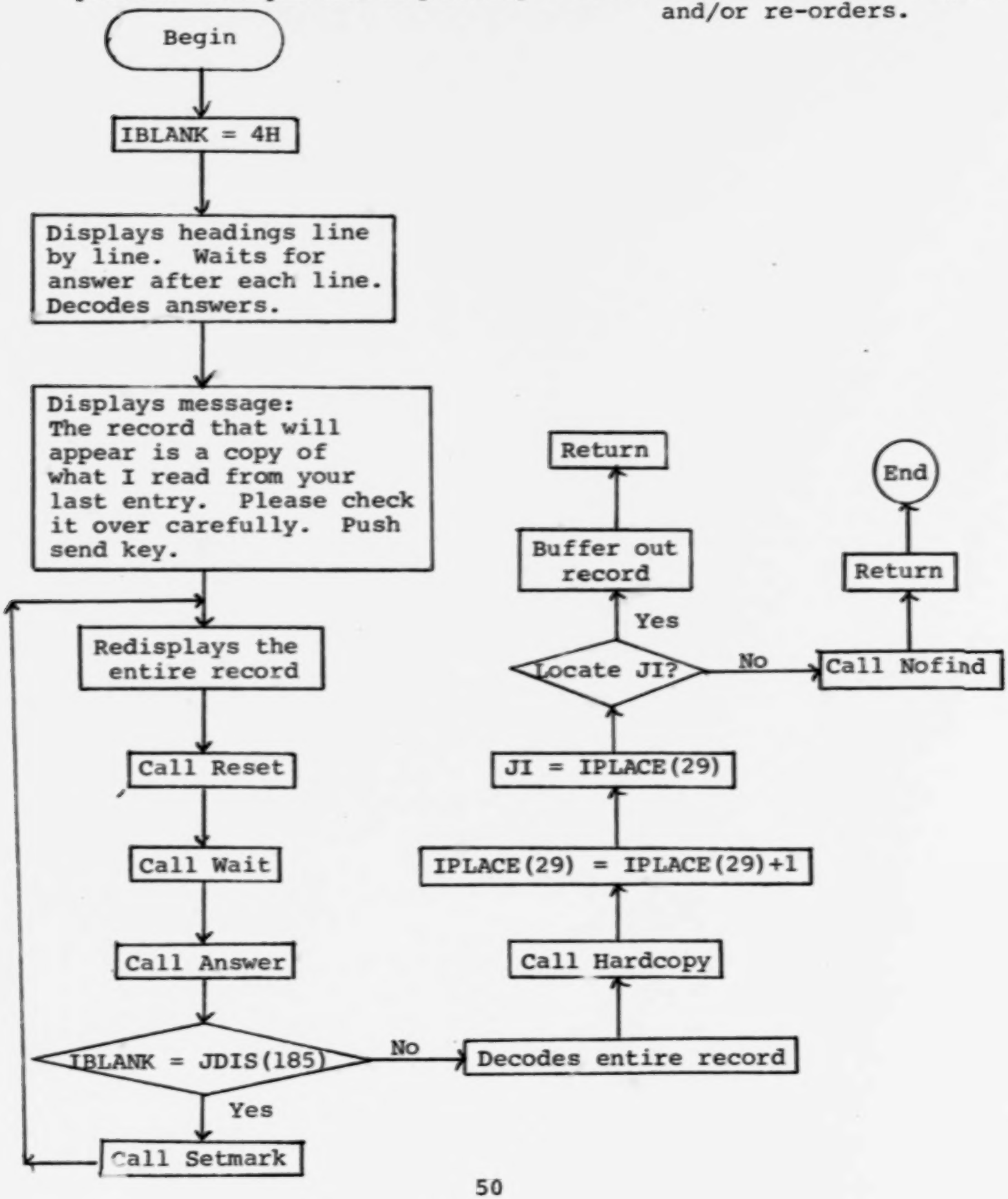


Variables

IDIS - memory display buffer
K - parameter, transfers number of choice to calling program

SUBROUTINE OSONEW

Operator types in new record, line by line. The record is redisplayed for any necessary changes. The computer checks to see if the marker is after the last piece of information. If it is, a hardcopy is made; if not, the computer redisplays the record previous to corrections. Circumvents the 2-phase process of input and requesting - could be used for rush requests and/or re-orders.

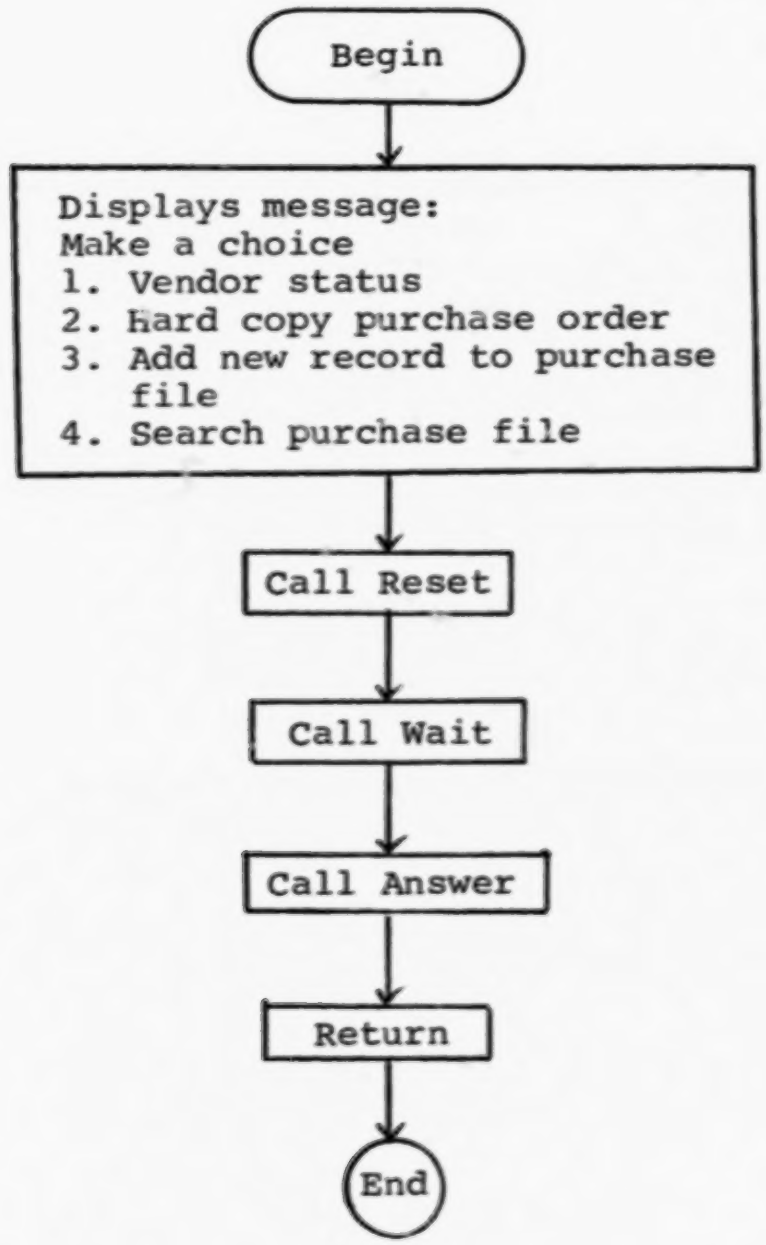


Variables

JI - location of last record on the file
IDIS - memory display buffer
JDIS - storage buffer
JDIS(185) - location of middle of order number in the record
   on the display screen
MESS - memory display buffer

SUBROUTINE PFILE(K)

Displays a message.

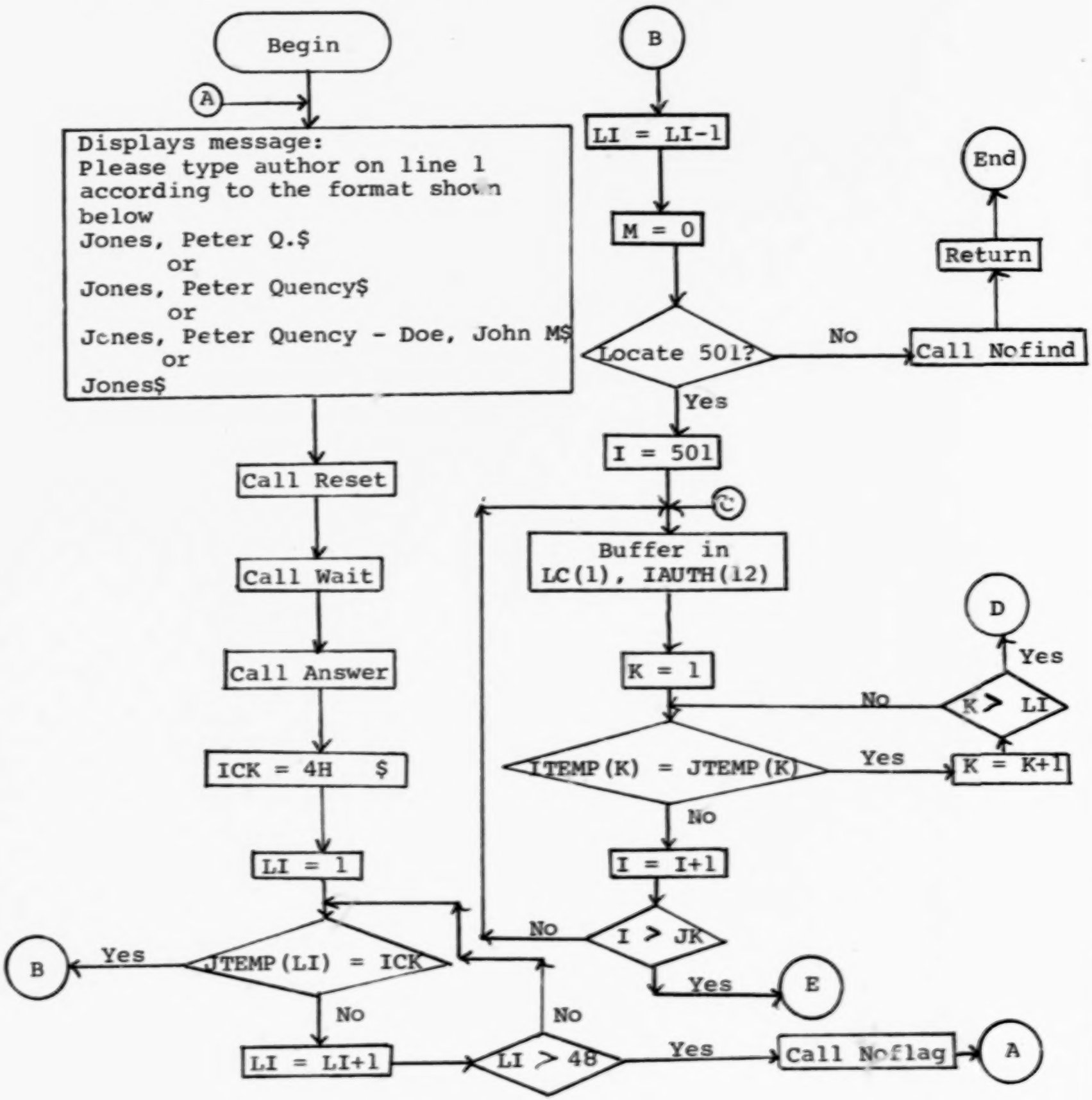


Variables

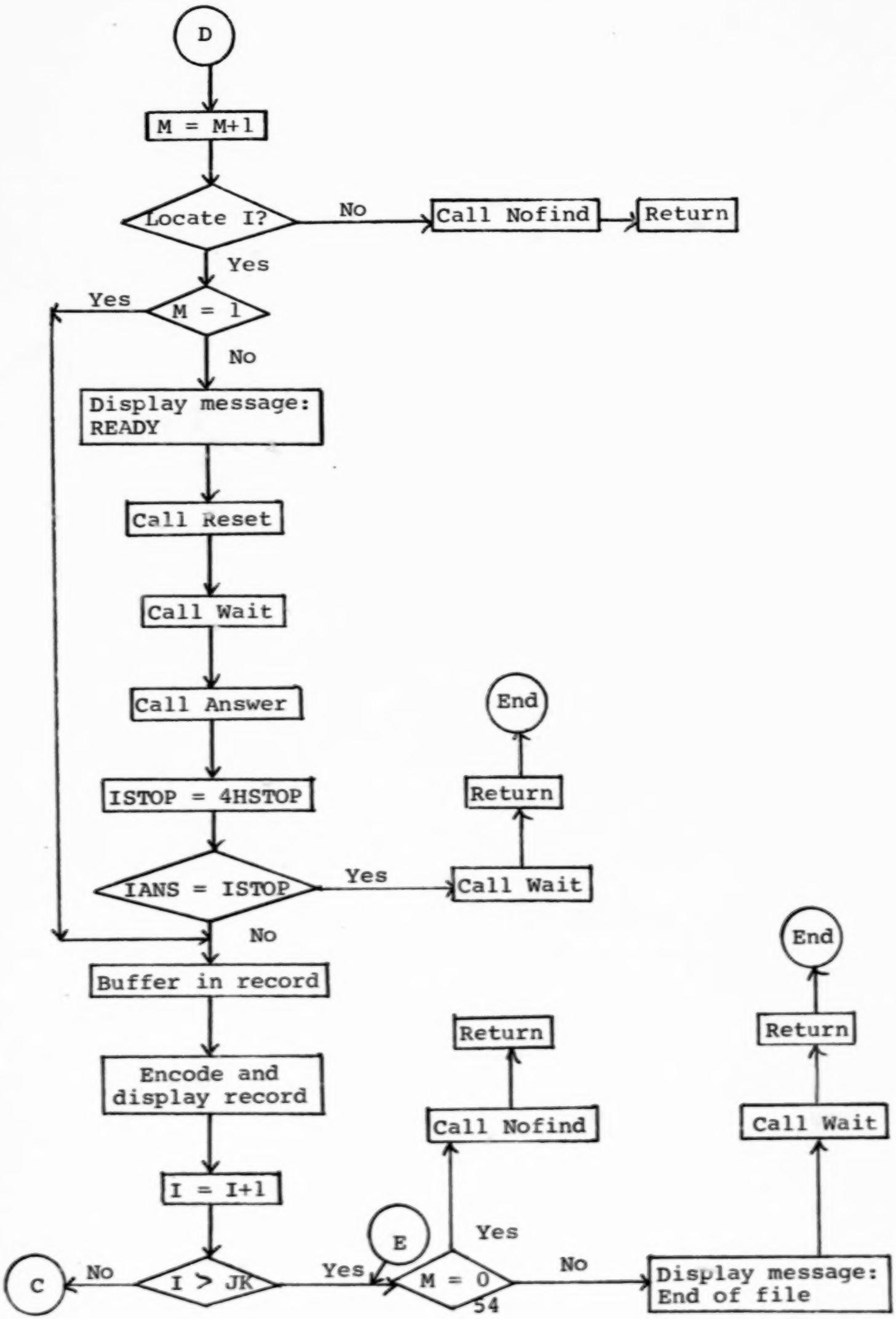
IDIS - memory display buffer
K - parameter, transfers number of choice to calling program

SUBROUTINE PSAUTH

User types in desired author. The computer searches for the first record with this name. While this record is being displayed, the computer continues to search for the next record with the same author. When it is found, READY is displayed on the screen. If the user wants the record that is already being displayed, he types STOP on the first line of the screen. However, if he wishes to continue searching, he presses the send key and the next record is displayed.



SUBROUTINE PSAUTH (Cont.)



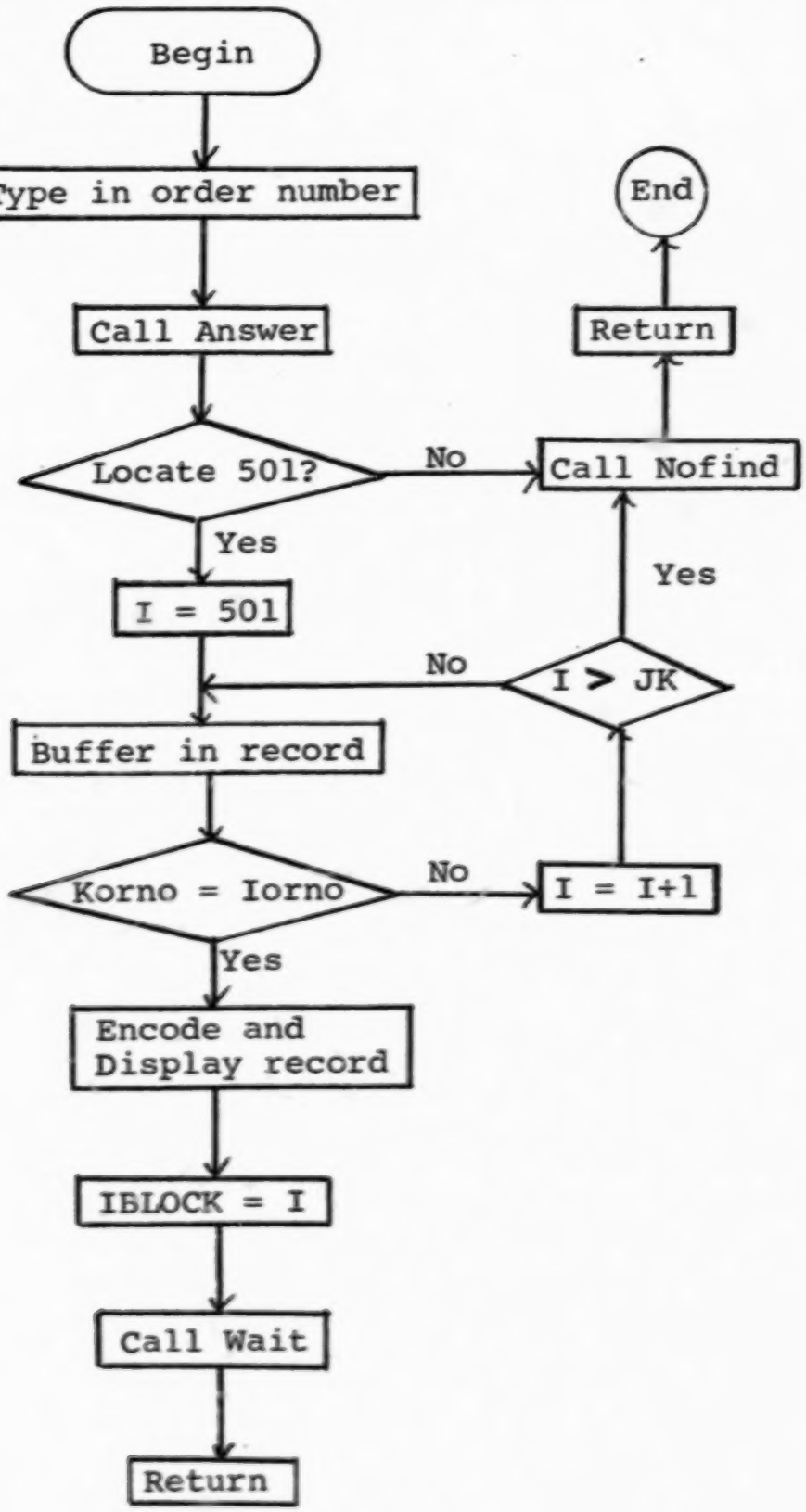
SUBROUTINE PSAUTH(cont.)

Variables

CHO - memory display buffer
JAUTH - storage buffer
LI - number of letters before the flag in an author's name
ICK - location of flag in memory (used as a testing device)
501 - location of first record in purchase file
JK - location of last record in purchase file
ITEMP(K) - equivalent to name of author on the record. However, its length varies according to the number of letters typed by the user
JTEMP(K) - name of author typed in by user
M - counter
IDIS - memory display buffer
IANS - storage buffer

SUBROUTINE PSORDER

Operator types in desired order number. The subroutine searches for it throughout the file. If and when it is found, the record is displayed.

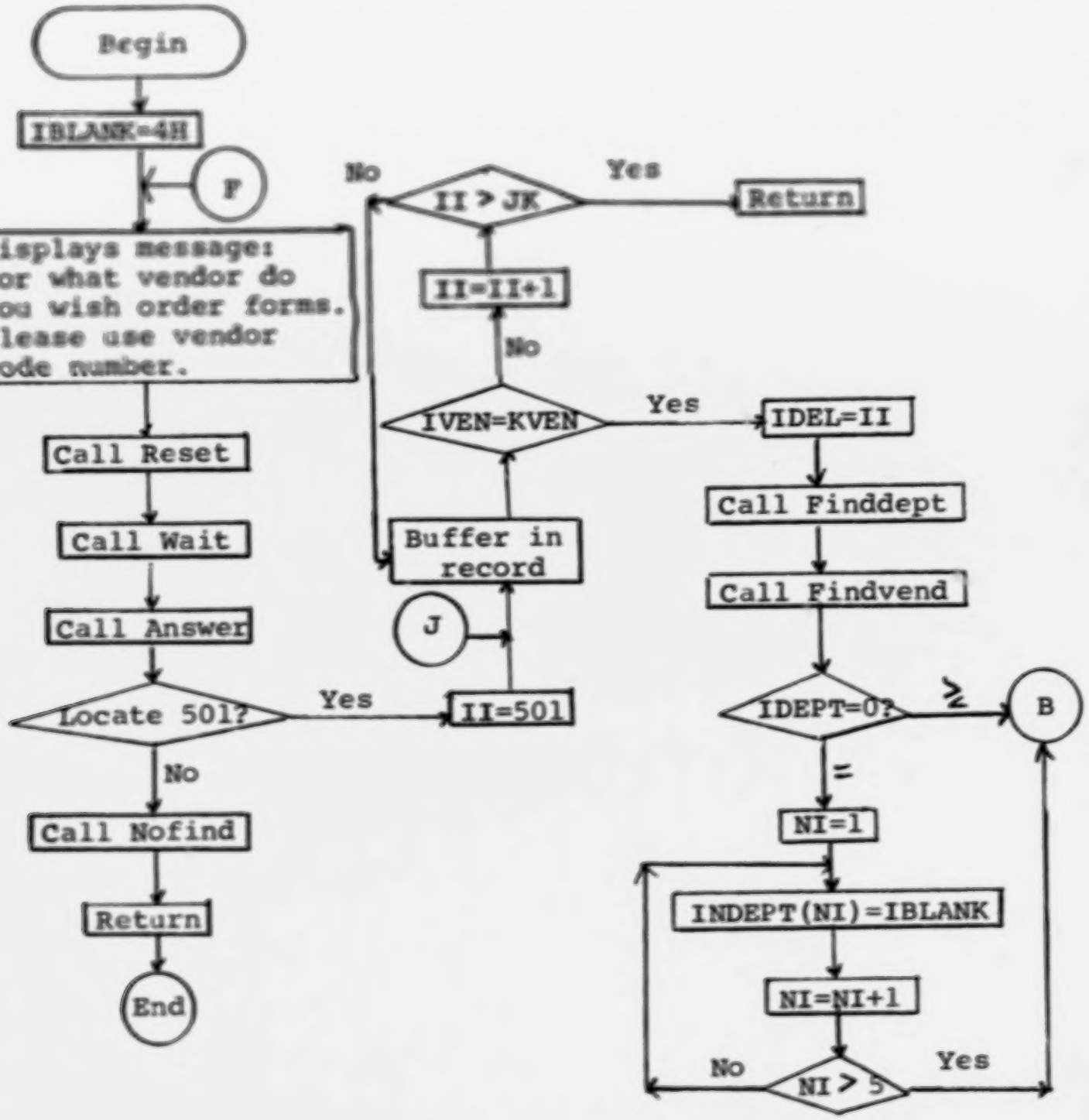


Variables

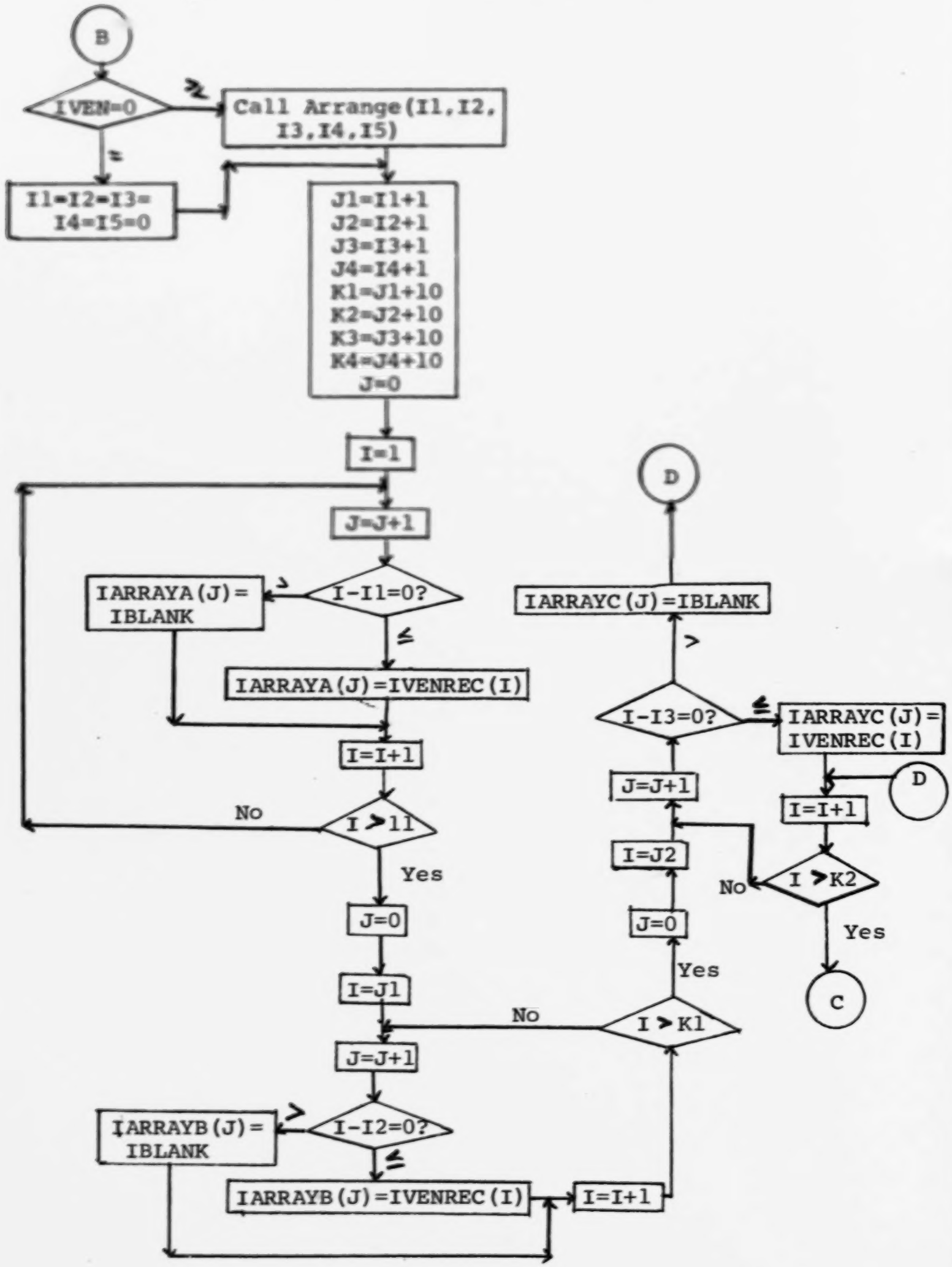
KORNO - location of order number typed in
JORNO - order number typed in
JK - location of last record on file
IORNO - order number on record
IDIS - memory display buffer
IBLOCK - location of displayed record

SUBROUTINE PURCHAS

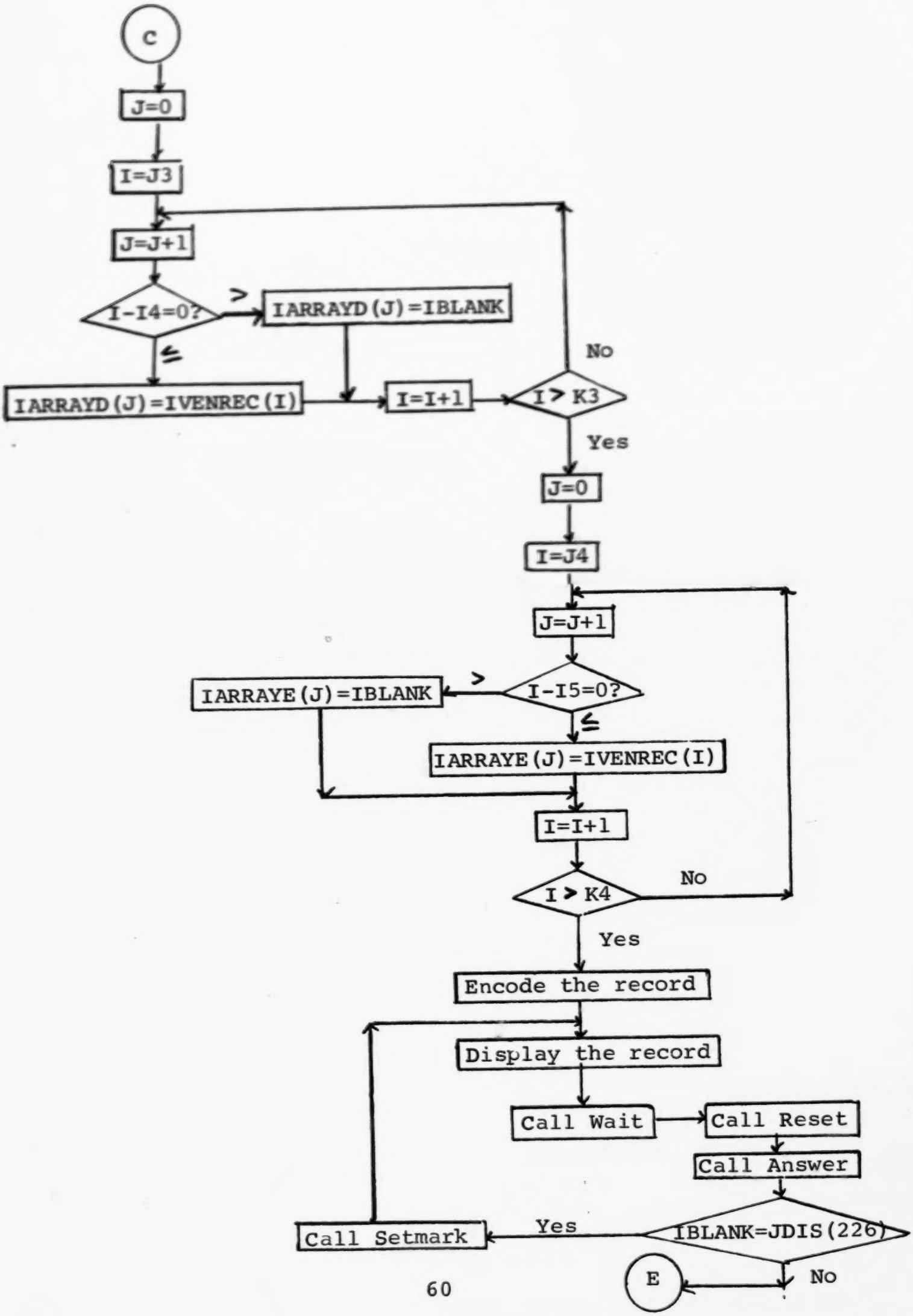
Displays a record requested by the user. The department code and vendor code are replaced by their name and name and address, respectively. If the displayed record is to be used for a purchase order, a hardcopy is produced, the record is deleted from the purchase file, and is transferred to the outstanding order file. If the displayed record is not desired, the searching process is repeated for a different vendor code. Order forms are produced for all records under a given vendor code.



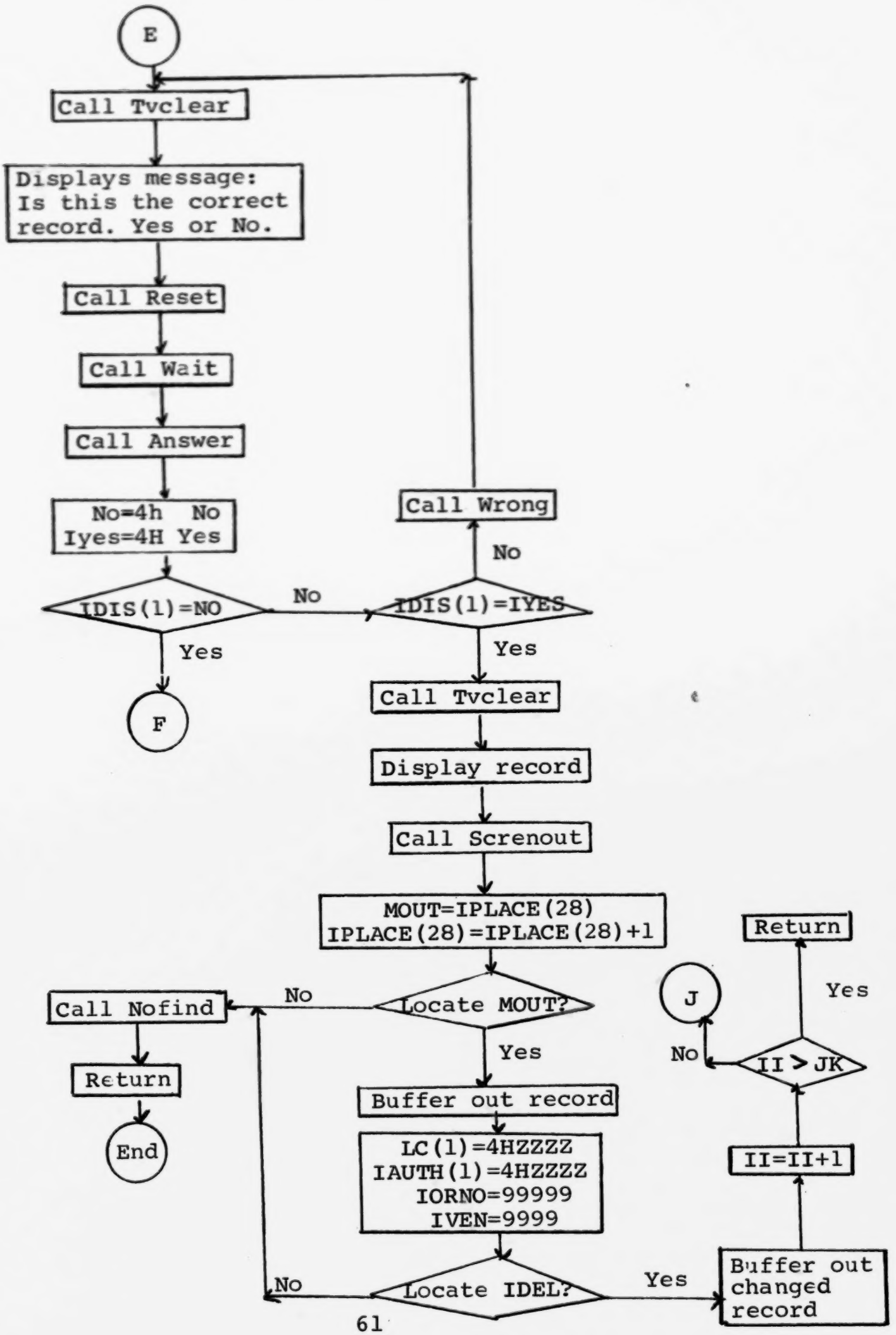
SUBROUTINE PURCHAS (cont.)



SUBROUTINE PURCHAS (cont.)



SUBROUTINE PURCHAS (cont.)

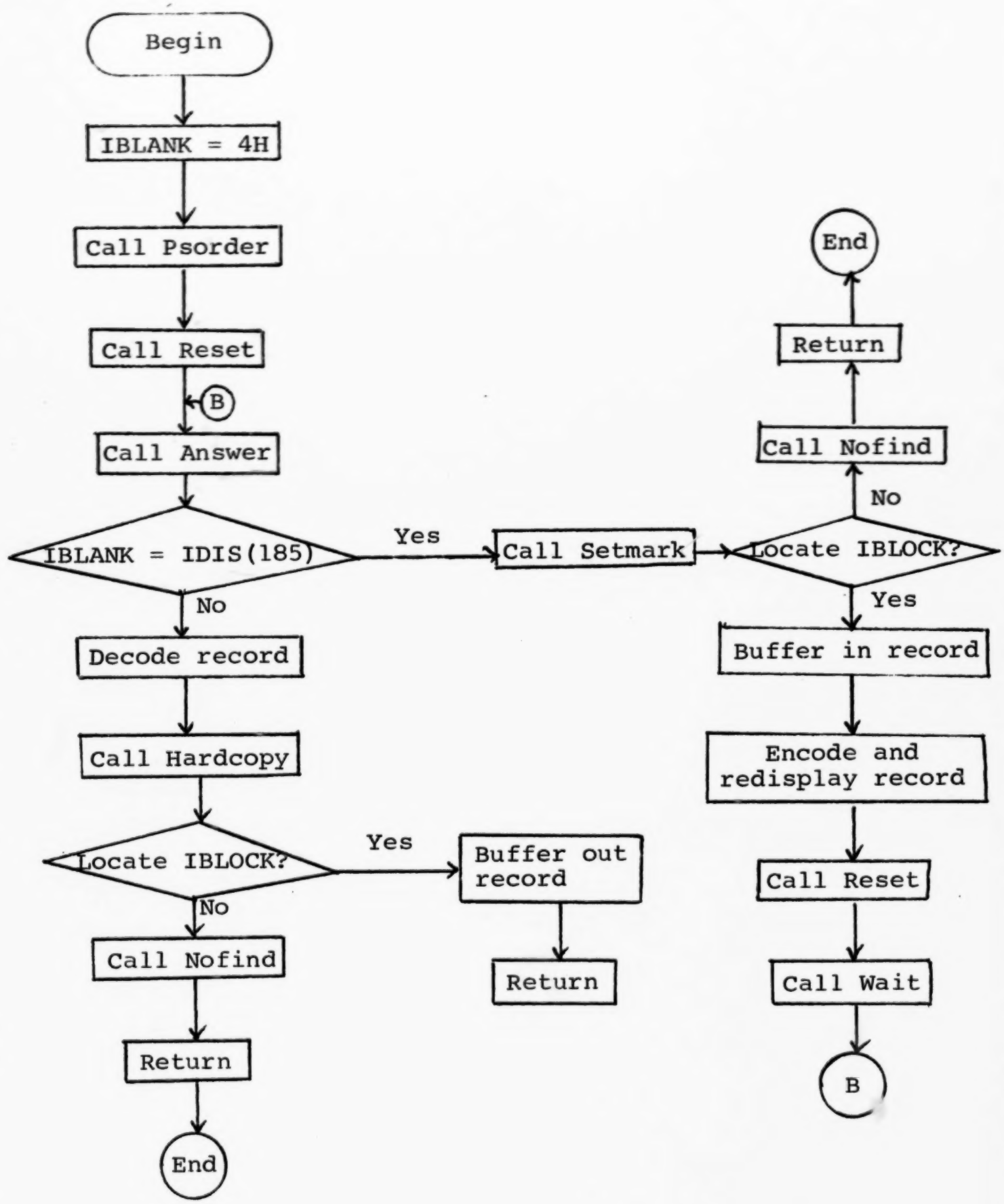


Variables

IBLANK - four blanks used as a testing device.
JDIS - memory display buffer
JVEN - vendor code typed in by user
KVEN - equivalent to JVEN
501 - location of first record in the purchase file
II - counter
JK - location of last record of purchase file
IVEN - vendor code on record in memory
IDEL - location of desired record
IDEPT - department code
NI - counter
INDEPT(NI) - array containing department name
I1=I2=I3=I4=I5=0 - parameters corresponding to the following variables: J1, J2, J3, J4, K1, K2, K3, K4 - variables designating the beginning and ending locations of each line of the vendor record.
IARRAYA(J),IARRAYB(J),IARRAYC(J),IARRAYD(J),IARRAYE(J) - line buffers for vendor record
IVENREC(I) - array containing the vendor record.
IDIS - memory display buffer
NO, IYES - testors
IPLACE(28) - location of last record of Z-block on outstanding order file.
MOUT - location of last record of new records on the outstanding order file.

SUBROUTINE PURCRDT

Displays record, checks to see if the marker is after the last piece of information. If it is not, the record is redisplayed; if it is, the record is decoded and a hardcopy is produced.

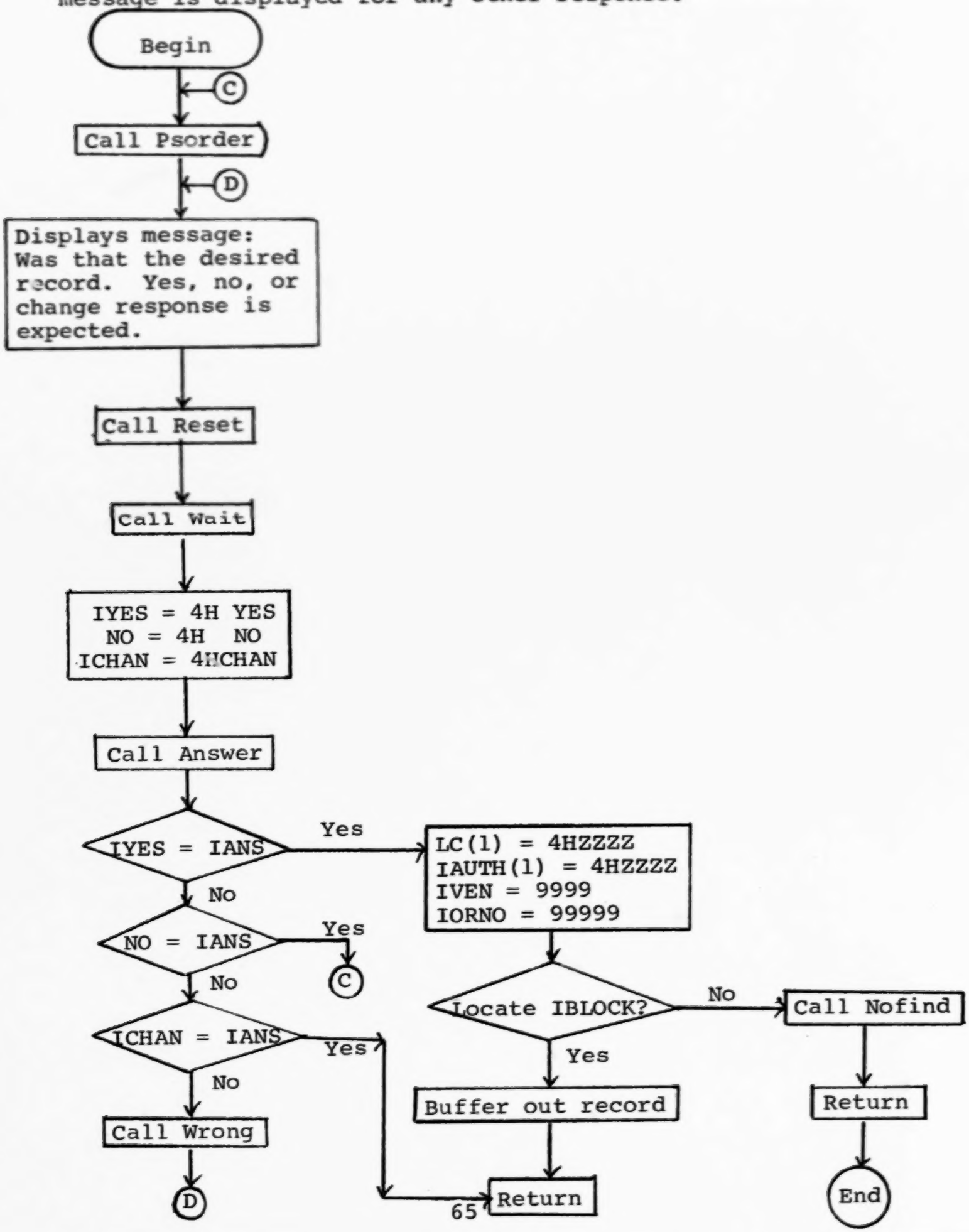


Variables

IDIS - storage buffer
IDIS(185) - location of middle of order number in the record being displayed
IBLOCK - location of desired record
IDIS - memory display buffer

SUBROUTINE PURDELT

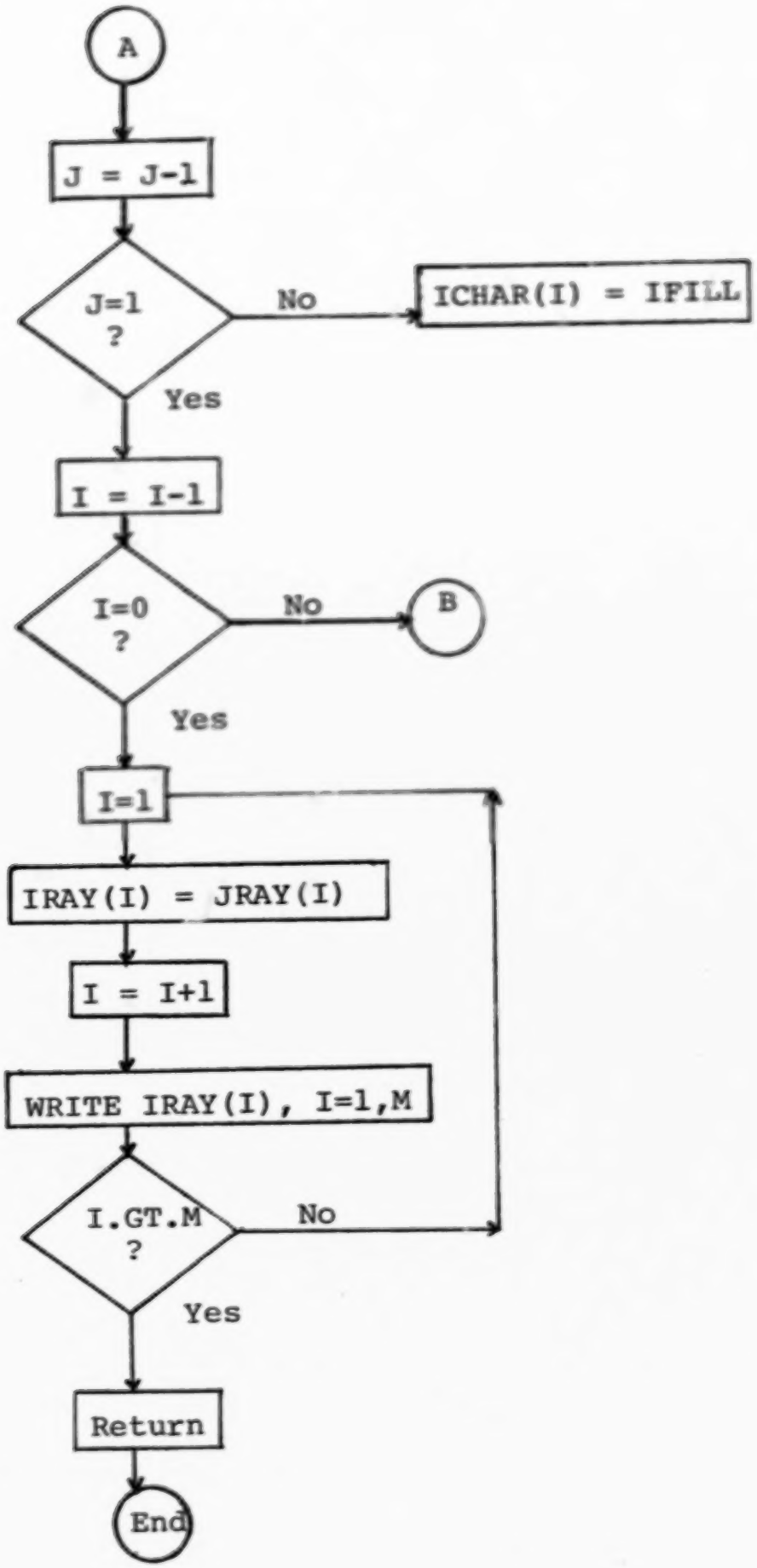
Operator types in a yes, no, or change response with regard to the displayed record. If the response is yes, the record is deleted; if it is no, subroutine Psorder is called; if it is change, the system is returned to FILECH; an error message is displayed for any other response.



Variables

IRES - memory display buffer
IANS - storage buffer
IBLOCK - block location of desired record

SUBROUTINE RJANSWER (cont.)

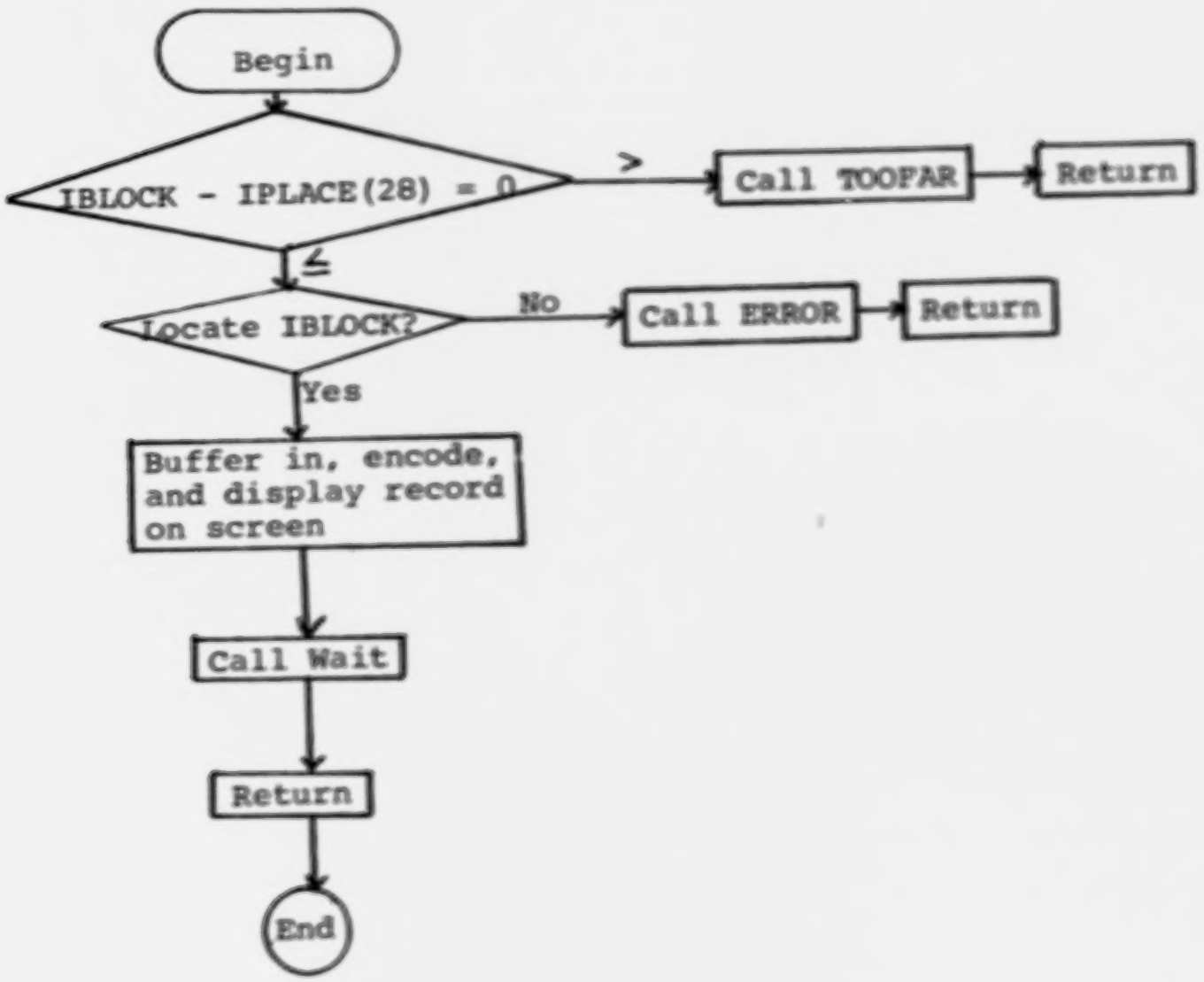


Variables

IRAY - word array output
ICHAR - character array
IDUMB - dummy array
IFILL - value determining zero or blank fill
N - number of characters
M - number of computer words
JRAY - word array input

SUBROUTINE RECORD

Checks to see if the operator has requested a record beyond the end of the file. If so, it calls TOOFAR; if not, it displays the record.

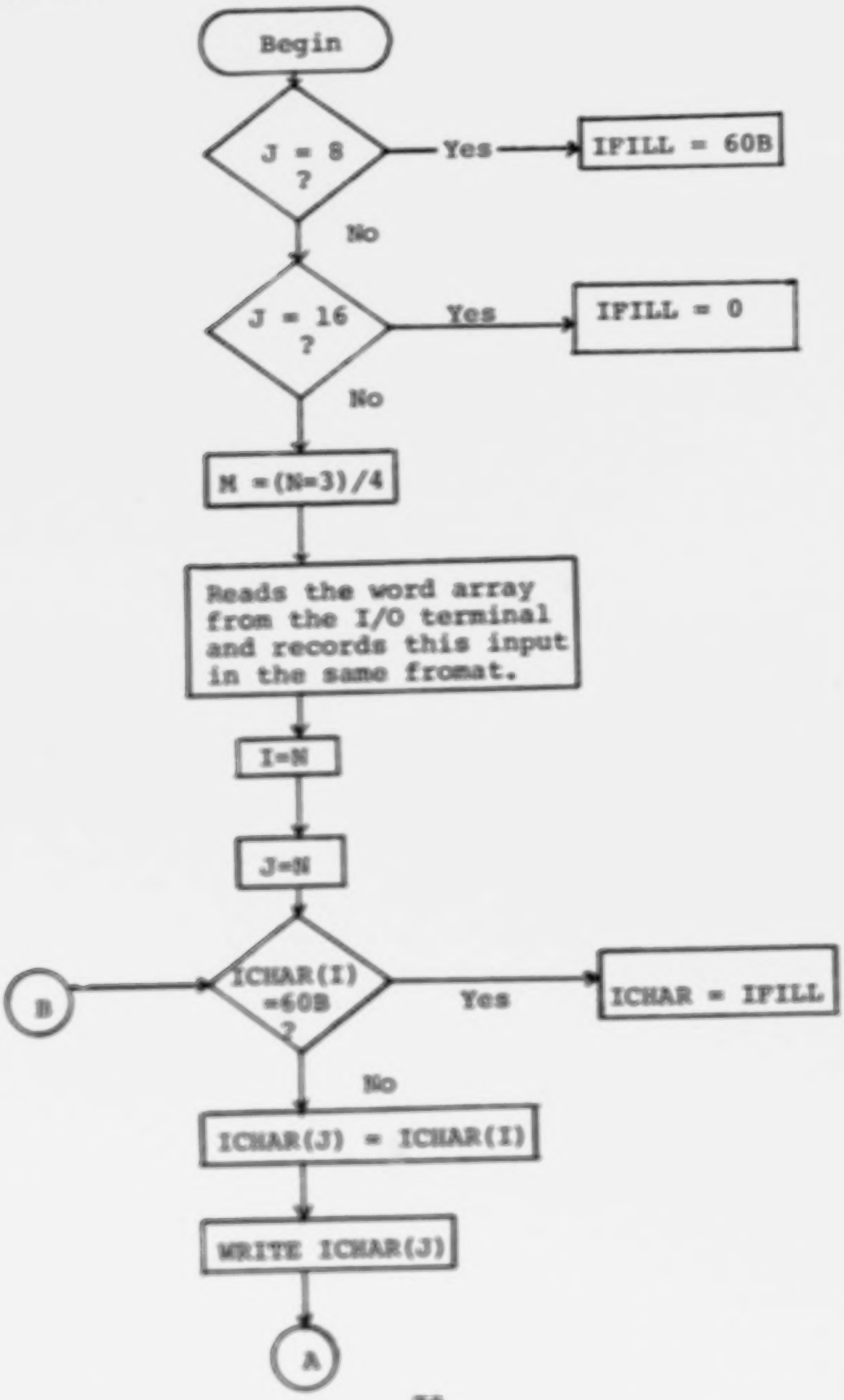


Variables

IBLOCK - location of desired record
IPLACE(28) - location of last record on file
LD - memory display buffer
IPLACE array - location of end record of each alphabetical block. However, IPLACE(1) = 1, is the first location of the file.

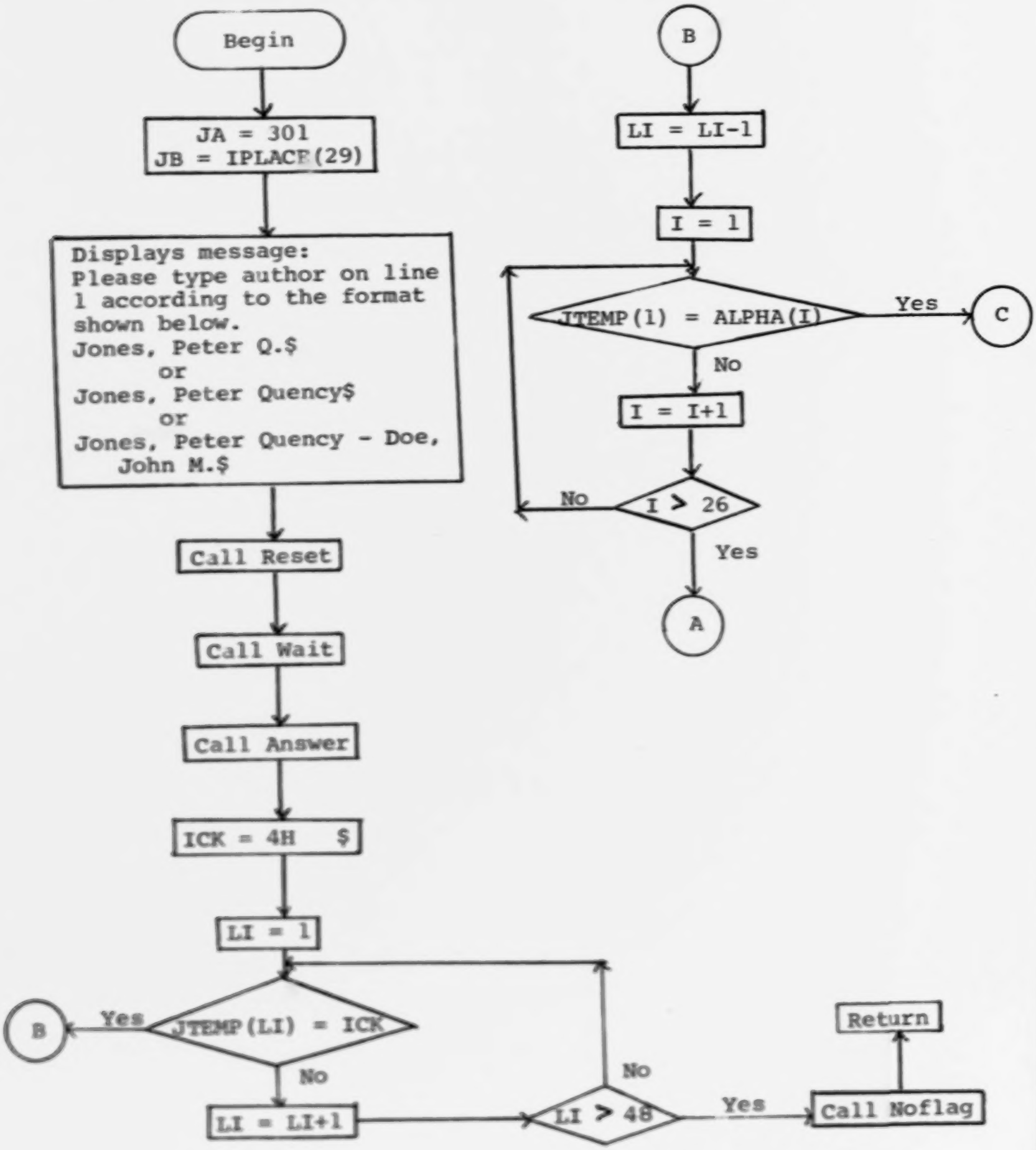
SUBROUTINE RJANSWER

Reads characters off the CRT screen, and, depending upon the format, right-justifies the characters, and zero (0) or blank-fills the stored representation for output into a fixed field.

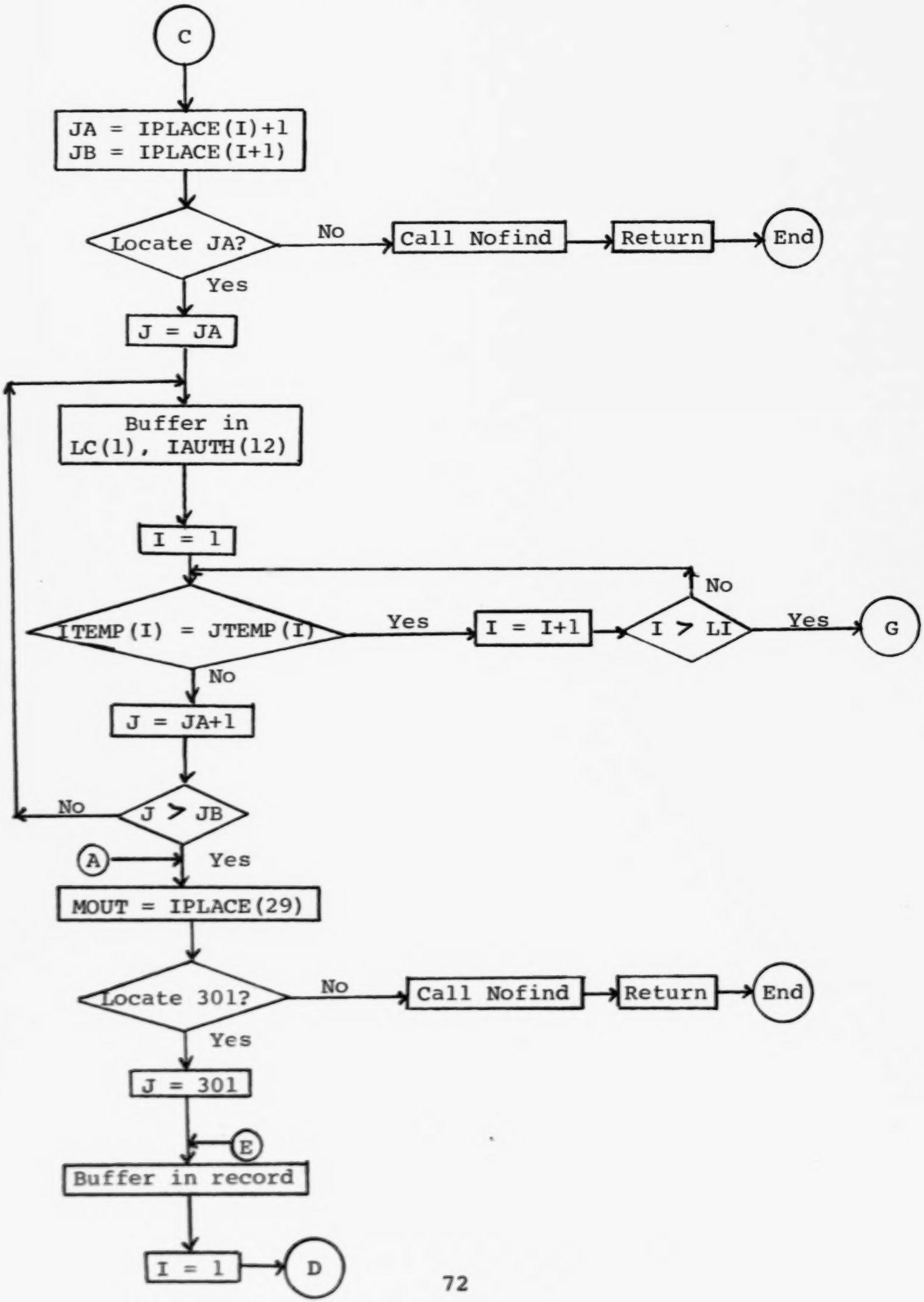


SUBROUTINE SAUTH

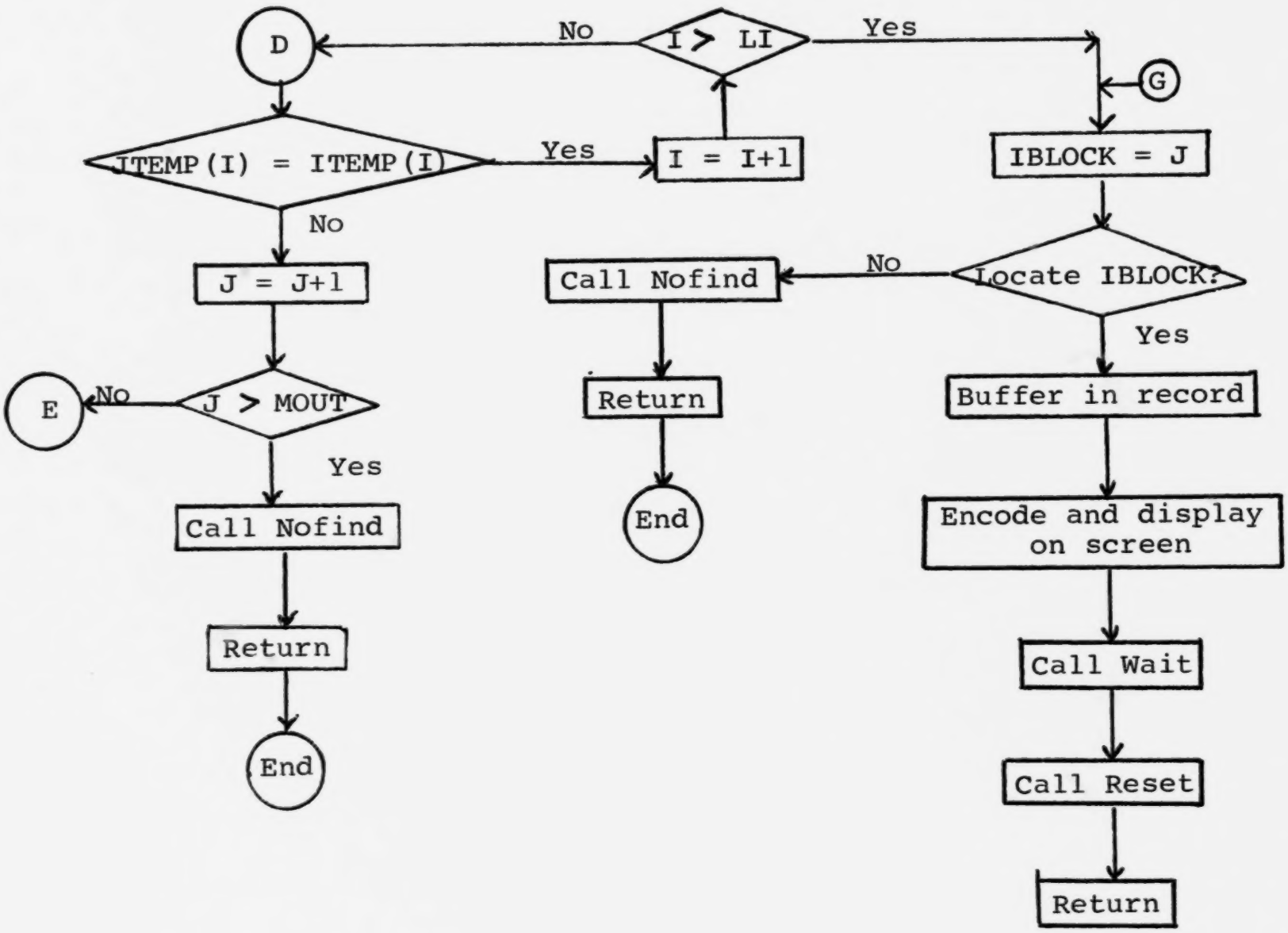
Searches for a particular author which has been typed in by the operator. If the record with this author is found, it is displayed. If it is not found, a Call Nofind message is displayed.



SUBROUTINE SAUTH (cont.)



SUBROUTINE SAUTH (cont.)

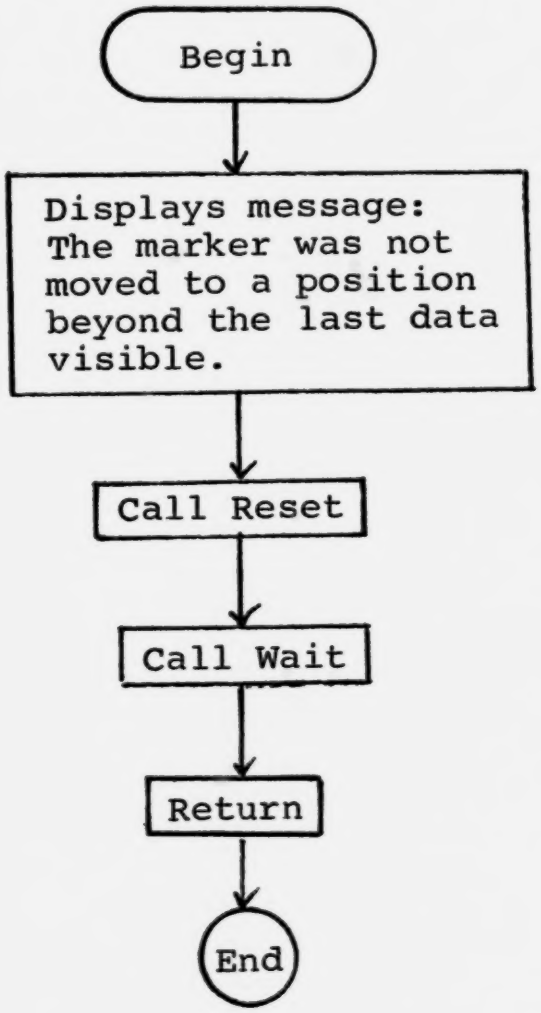


Variables

CHO - memory display buffer
JAUTH - storage buffer
ICK - location of flag in memory (used as a testing device)
LI - number of letters before the flag in an author's name
JTEMP(1) - first letter of author's name typed by the user
ALPHA (I) - alphabetical array, when I = 1, equals A-block
JA = IPLACE(I)+1 - location of first record in desired alphabetical block
JB = IPLACE(I+1) - location of last record in desired alphabetical block
MOUT = IPLACE(29) - location of last record of new records
ITEMP(I) - equivalent to name of author on the record. However, its length varies according to the number of letters typed by the user
IBLOCK - location of desired record
IDIS - memory display buffer
301 - location of first record of new records
IPLACE(29) - location of last record of new records

SUBROUTINE SETMARK

Displays a message.

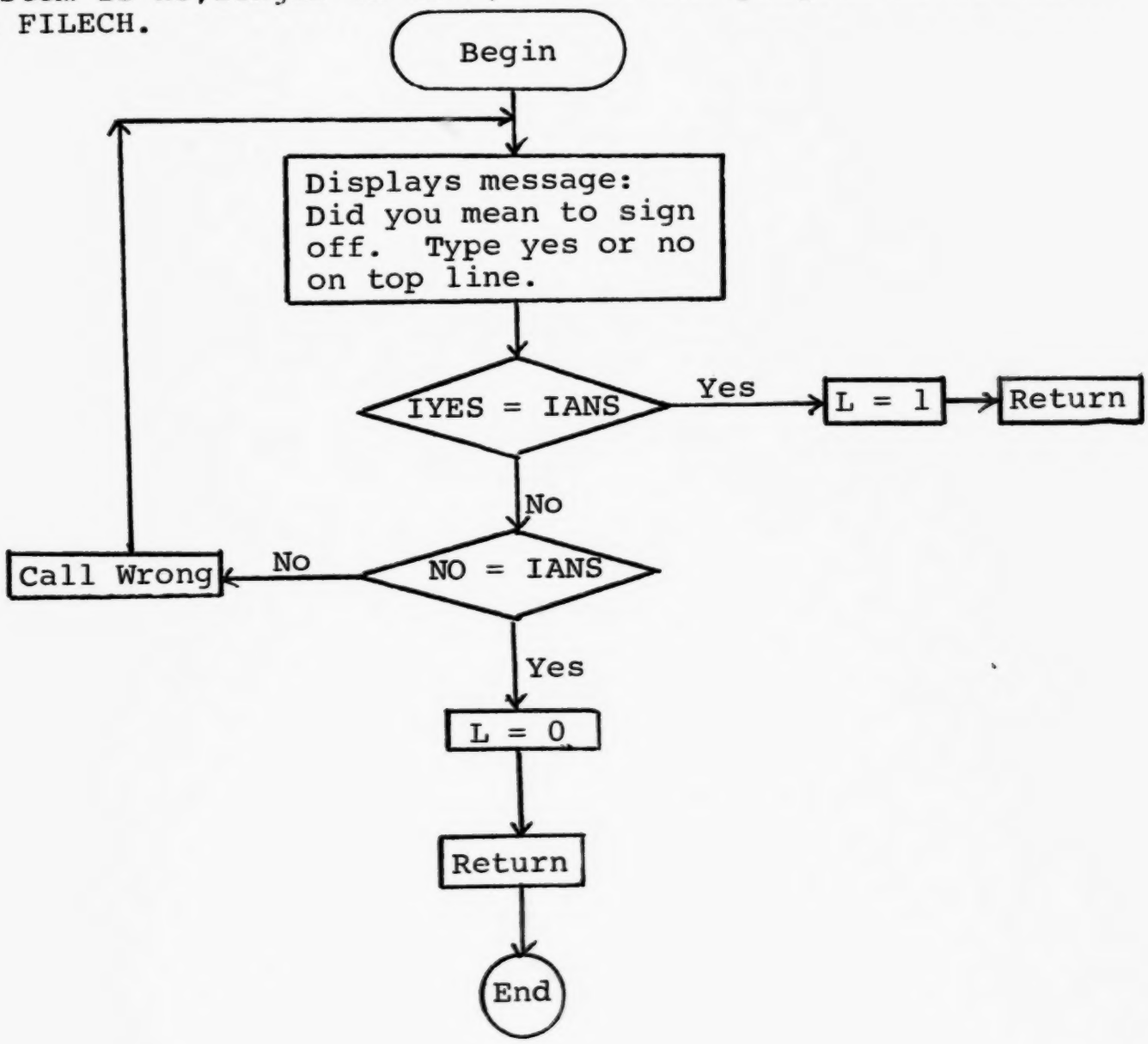


Variables

IRES - memory display buffer

SUBROUTINE SIGNOFF(L)

Checks to see if operator really wants to sign off. The parameter, L, transfers the response, yes or no, to the calling program. If L=1 the last page is displayed and the system is no,longer on-line, if L=0 the program is returned to FILECH.

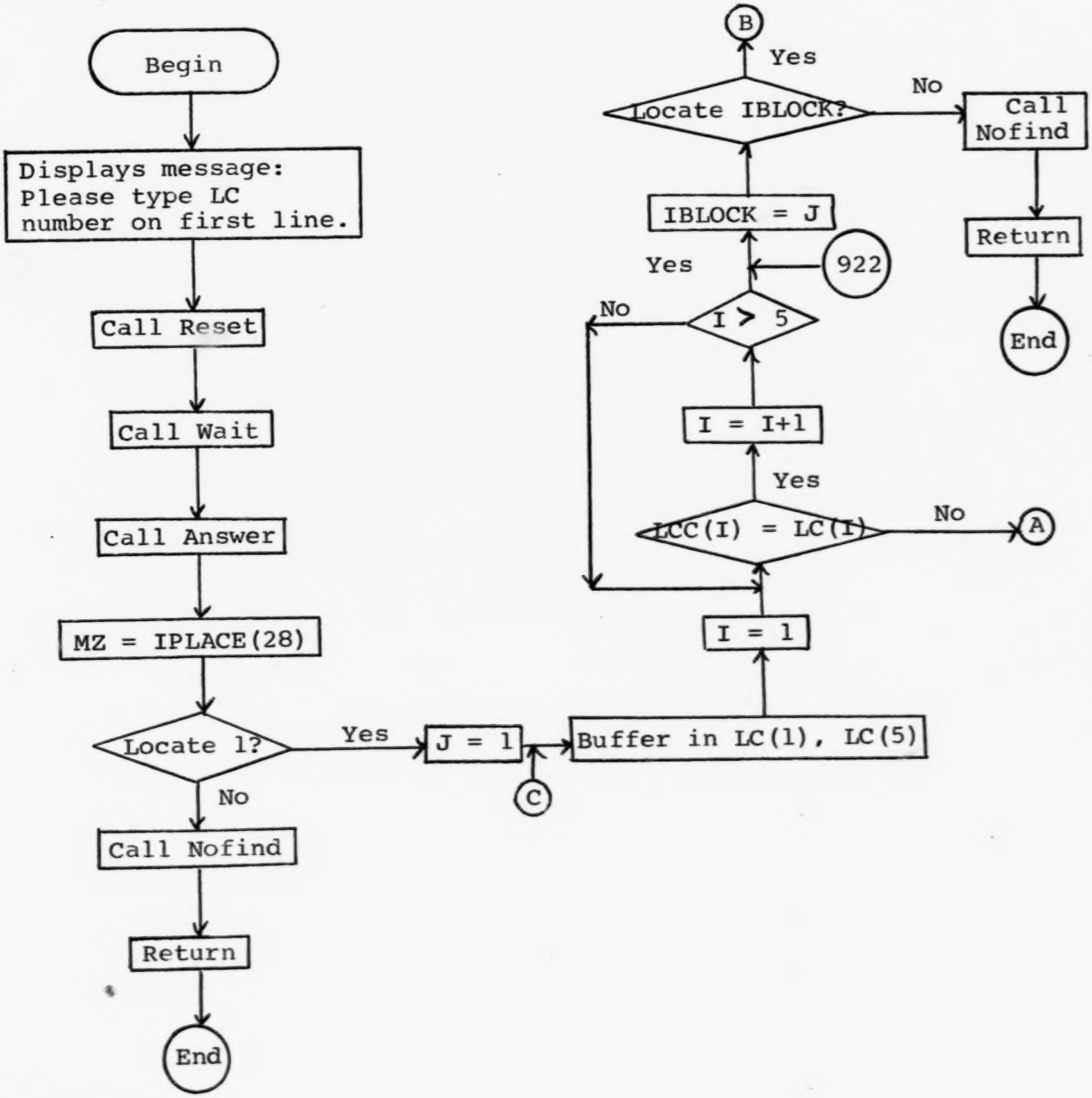


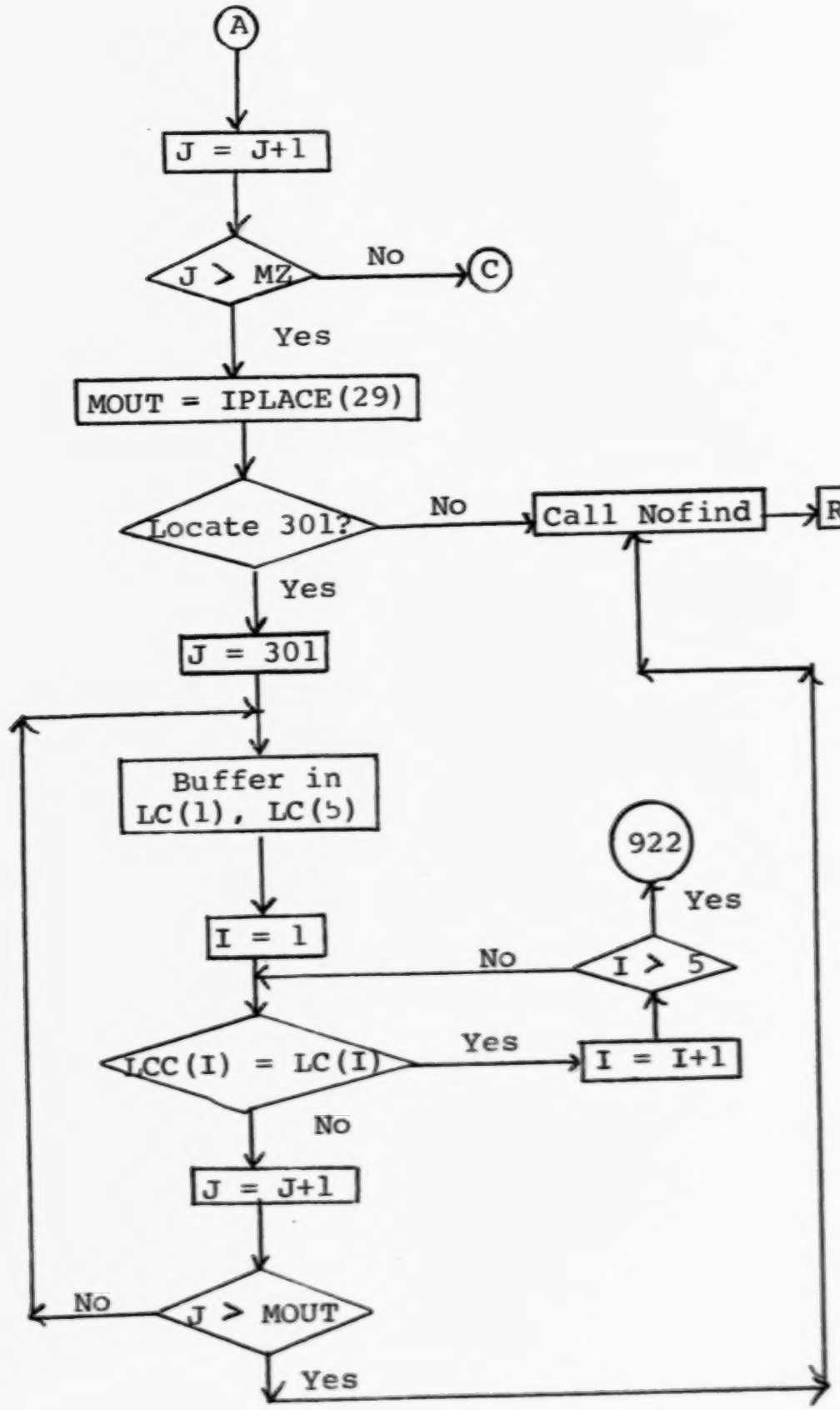
Variables

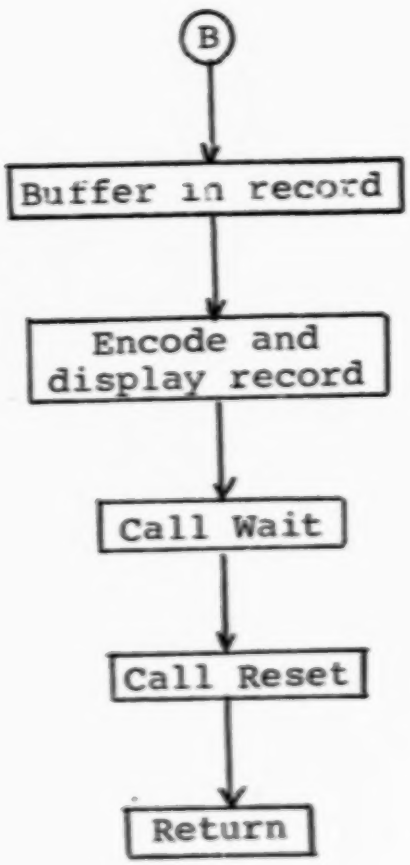
IRESP - memory display buffer
IANS - storage buffer
L - parameter used to transfer the yes or no response.

SUBROUTINE SLC

The user types in the desired LC number. The computer searches through the entire file, including new records, until the record is found. If it is found, it is displayed; if not, a call NOfind message is displayed.




A
J = J+1
J > MZ
No
C
Yes
MOUT = IPLACE(29)
Locate 301?
No
Call Nofind
Return
End
Yes
J = 301
Buffer in
LC(1), LC(5)
922
Yes
I = 1
No
I > 5
LCC(I) = LC(I)
Yes
I = I+1
No
J = J+1
No
J > MOUT
Yes


B
Buffer in record
Encode and
display record
Call Wait
Call Reset
Return

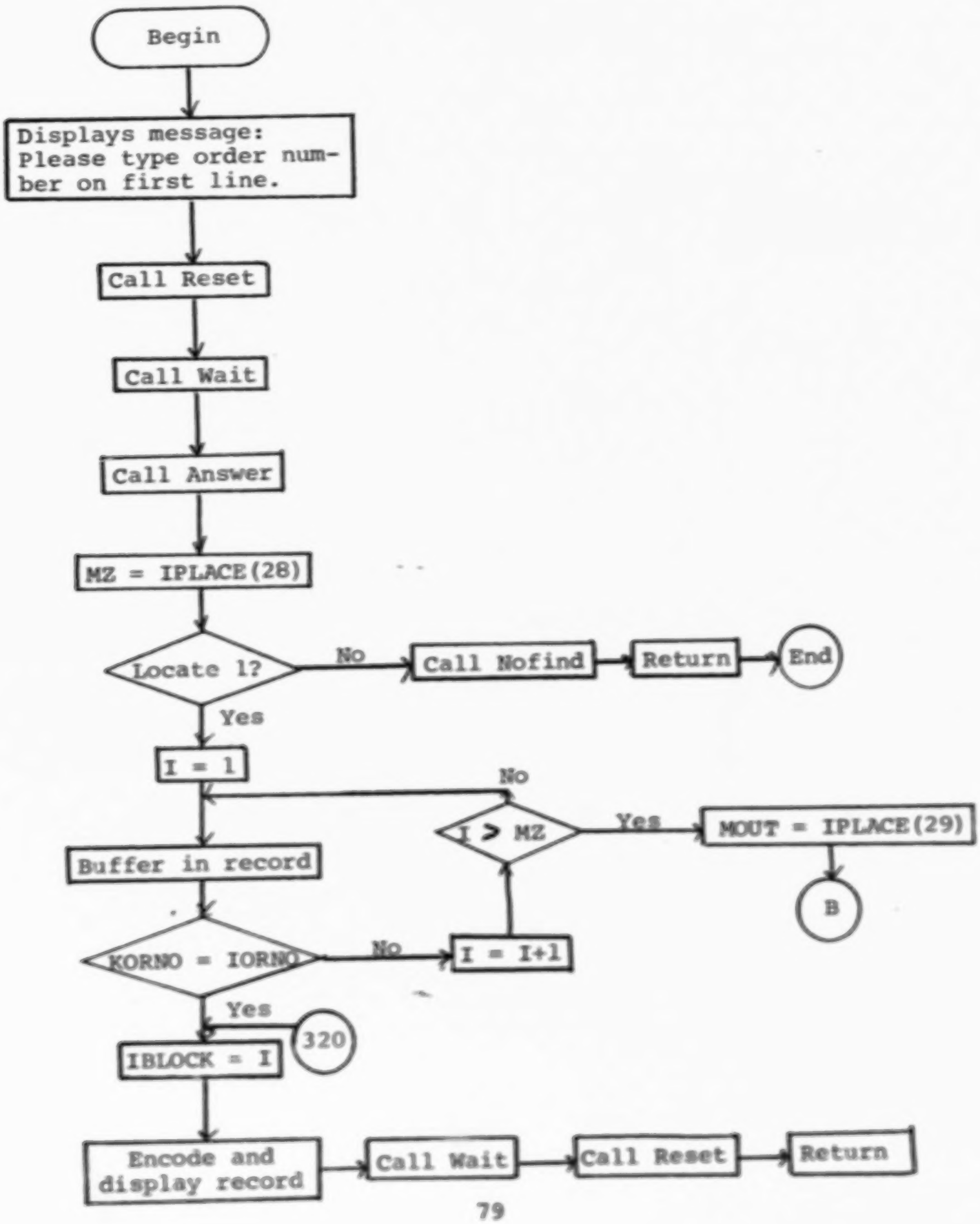
SUBROUTINE SLC (cont.)

Variables

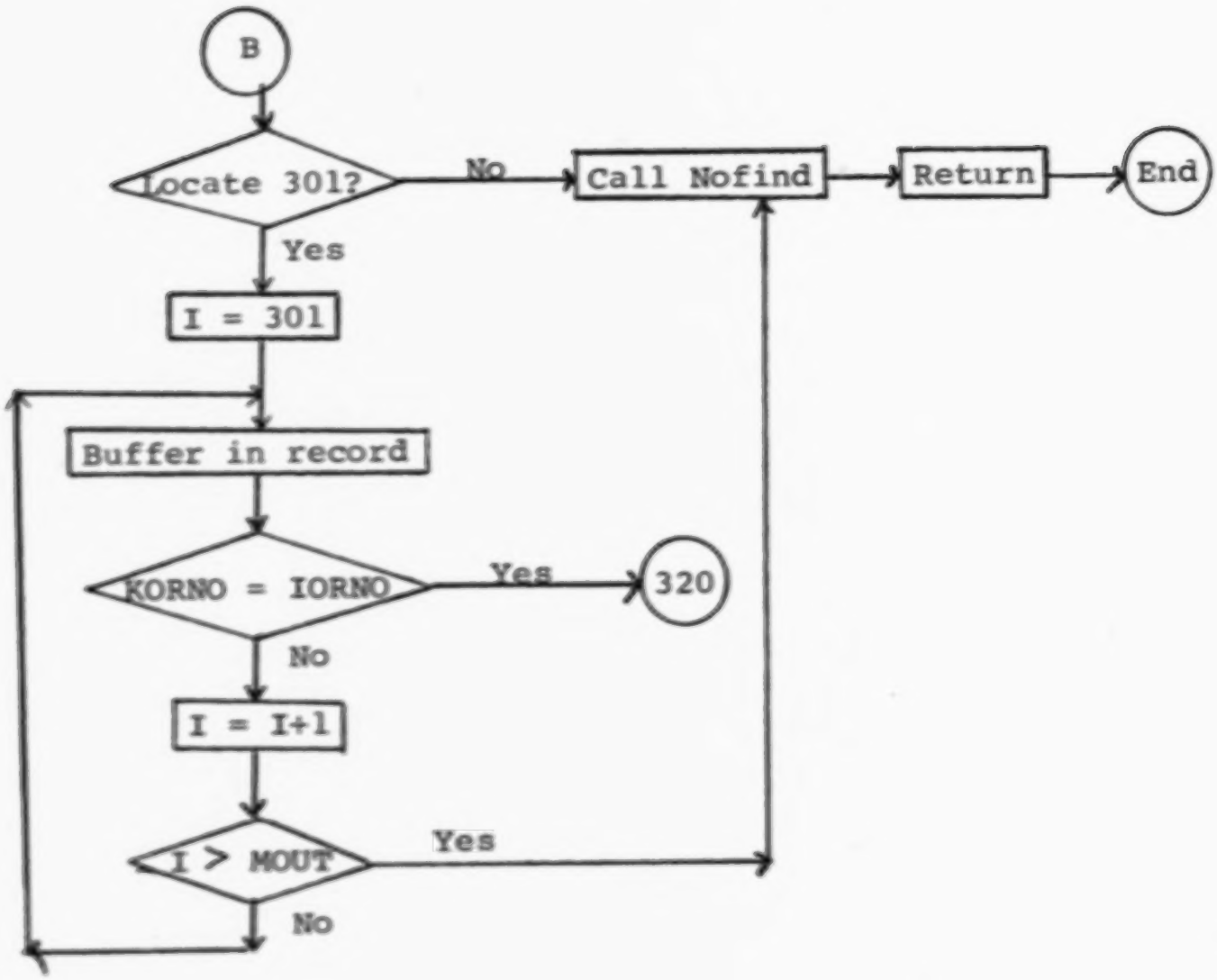
CHO - memory display buffer
LCC - storage buffer
MZ = IPLACE(28) - location of last record of the Z-block
LC(I) - LC number in the record
LCC(I) - typed-in LC number
MOUT = IPLACE(29) - location of the last record of the new record
IBLOCK - location of desired record
IDIS - memory display buffer

SUBROUTINE SORDER

Operator types in desired order number. The computer searches through the file for the record with this number. If it is found, it is displayed. If it is not found, a Call Nofind message is displayed.



SUBROUTINE SORDER (cont.)

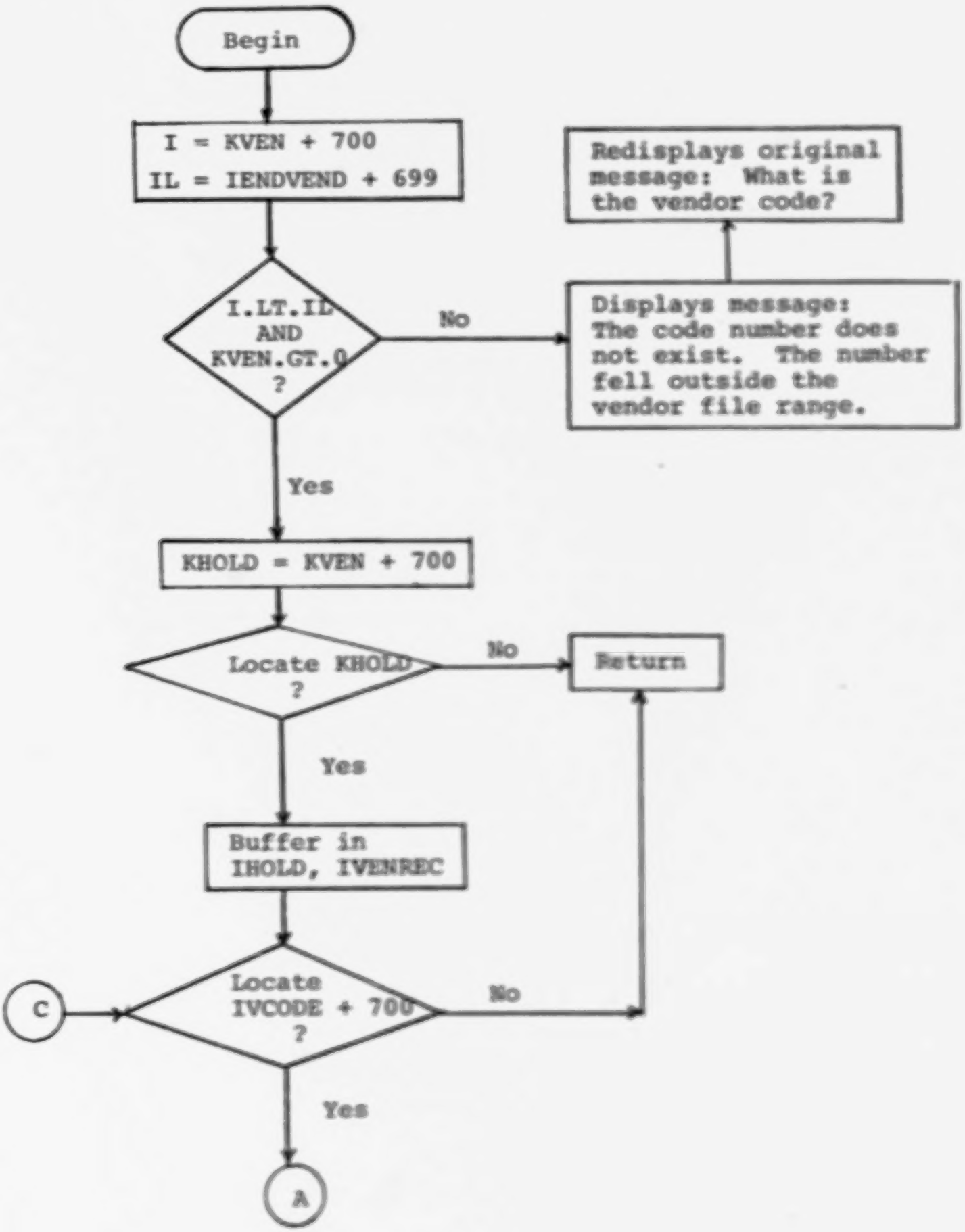


Variables

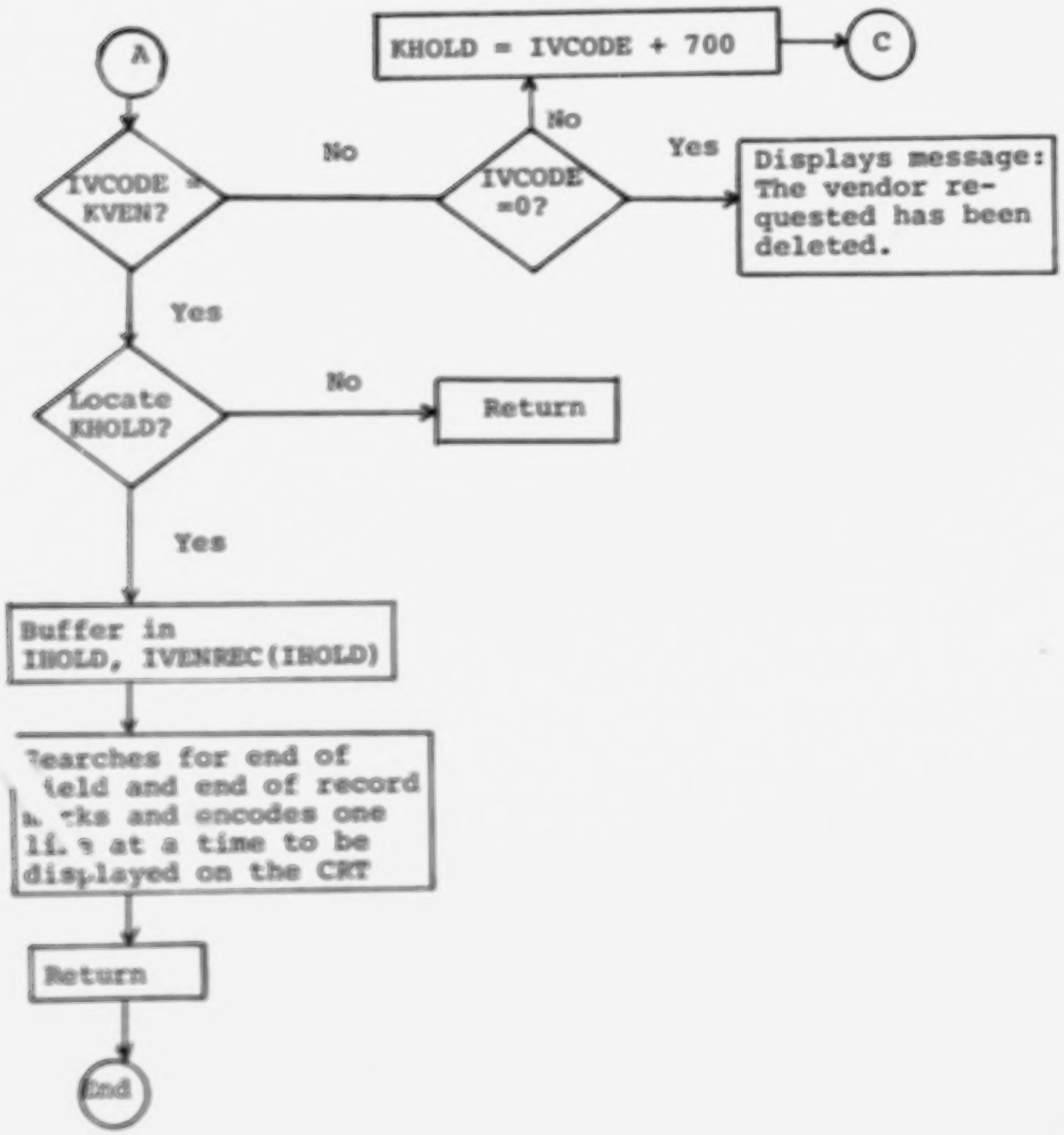
CHO - memory display buffer
JORNO - storage buffer
KORNO - location of typed-in order number
MZ = IPLACE(28) - location of last record of Z-block
IORNO - order number in displayed record
MOUT = IPLACE(29) - location of last record in the new records
301 - location of first record of new records
IDIS - memory display buffer
IBLOCK - location of desired record

SUBROUTINE SVCODE

Operator types in desired vendor code, on which he wishes to search. The system searches for the vendor record with that code. It is then found and displayed, or if the code does not exist, an information statement is displayed informing the user of this.



SUBROUTINE SVCODE (cont.)

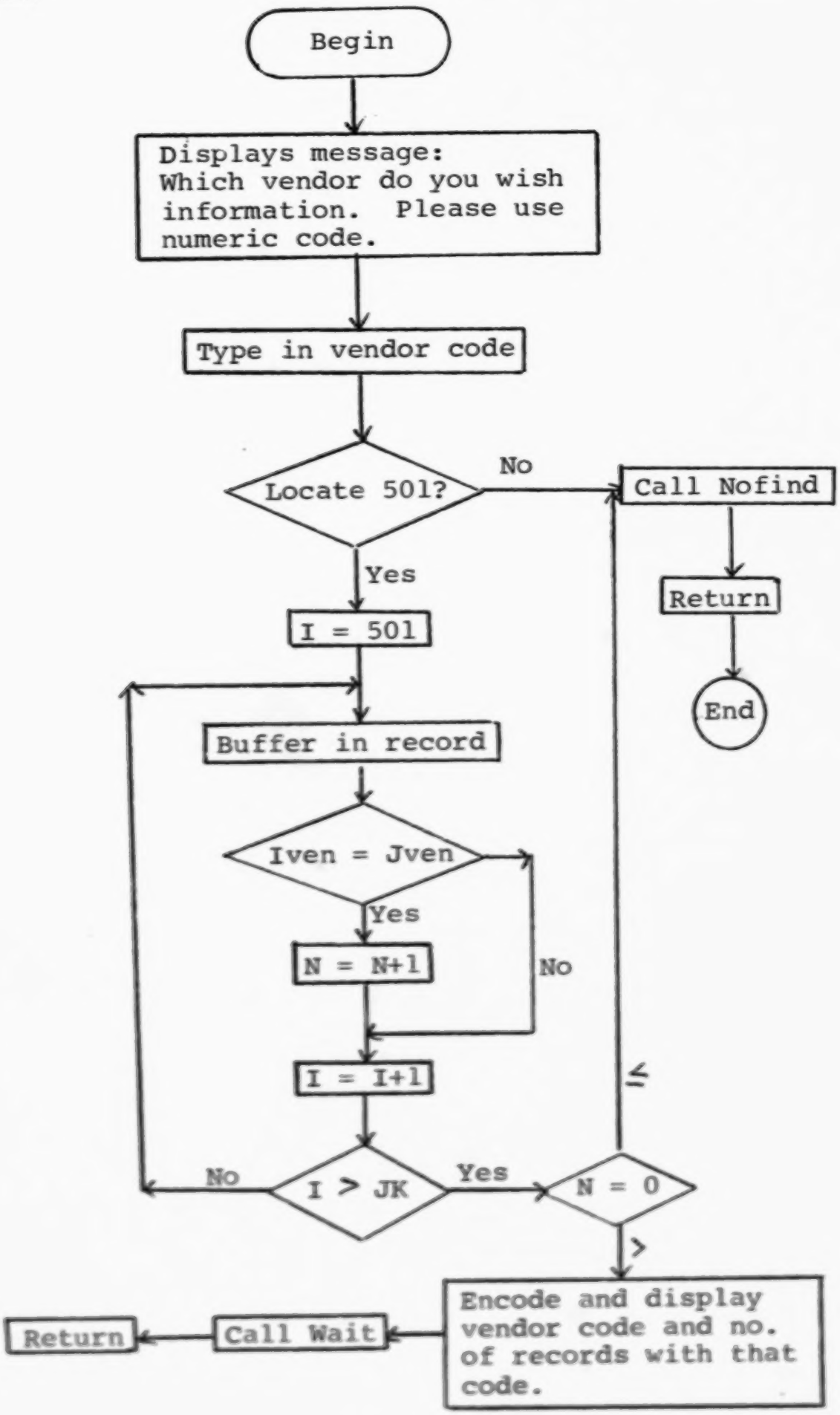


Variables

ITEMP - first letter of record
KTEMP - storage buffer
KVEN - code typed in by user
IVCODE - vendor code on record
IL - position before the first vendor record
IENDVEND -
IHOLD - length of record
KHOLD - storage location of vendor record
IVENREC - array containing vendor record

SUBROUTINE SVEND

Operator types in desired vendor code. The number of records with that code are counted and the total number is displayed.



Variables

N - record counter

IDIS - memory display buffer

KVEN - storage buffer

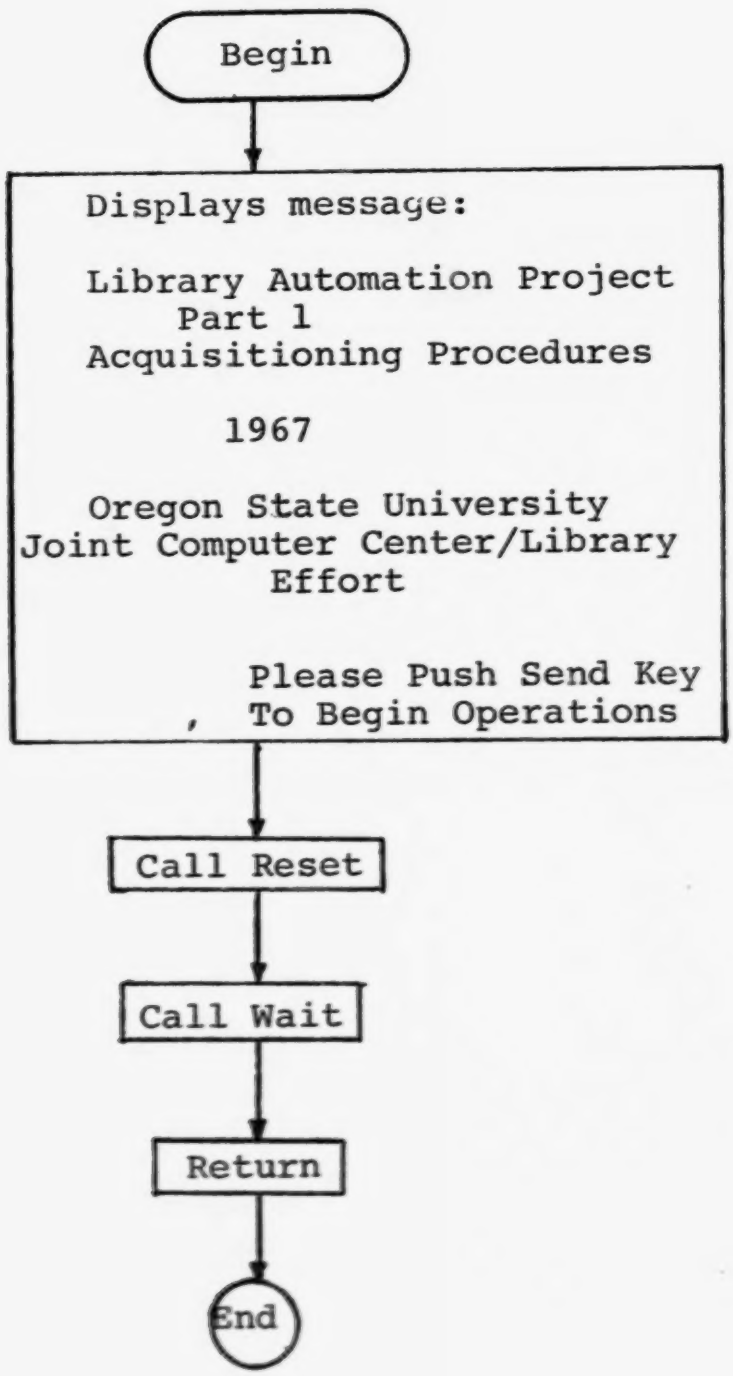
JVEN - memory location of typed-in vendor code

IVEN - vendor code on each record

IANS - memory display buffer

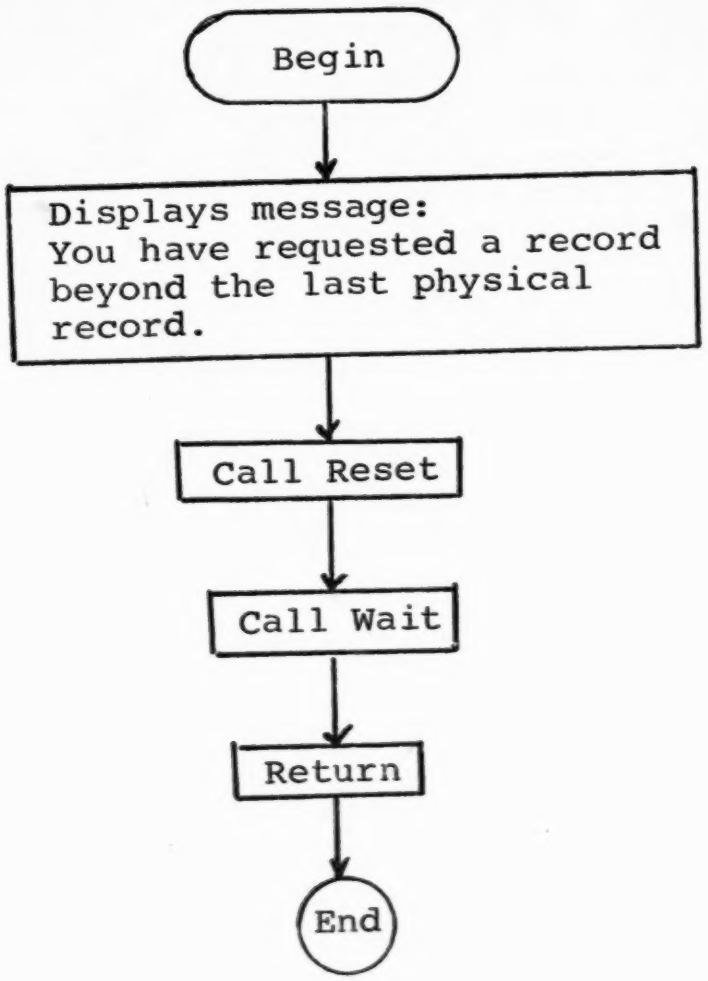
SUBROUTINE TITLPAGE

Displays a message



SUBROUTINE TOOFAR

Displays a message.

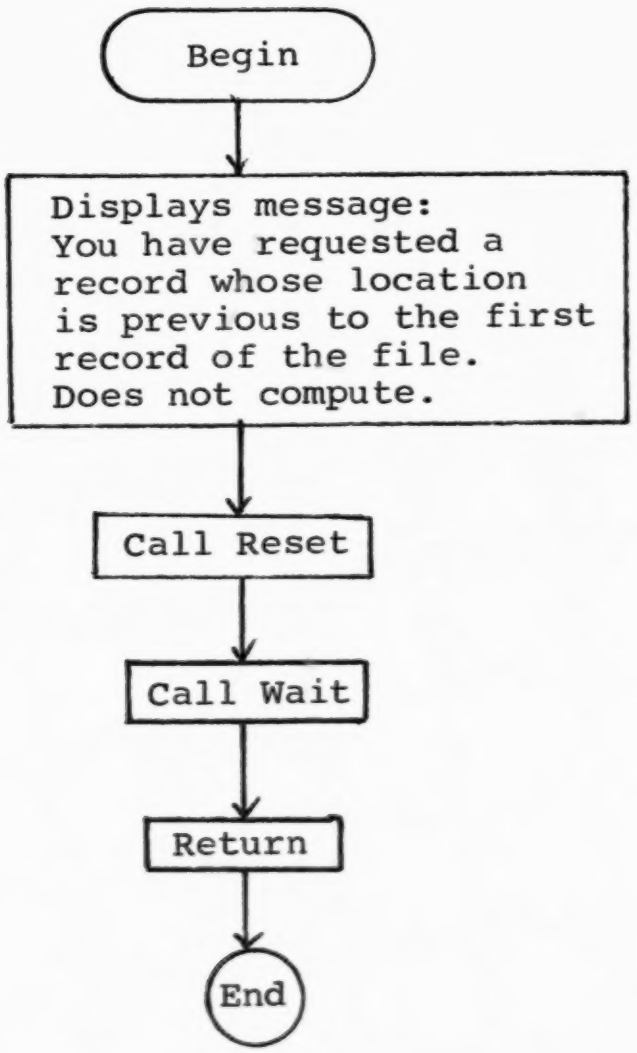


Variables

IANS - memory display buffer

SUBROUTINE TOOSMALL

Displays a message.

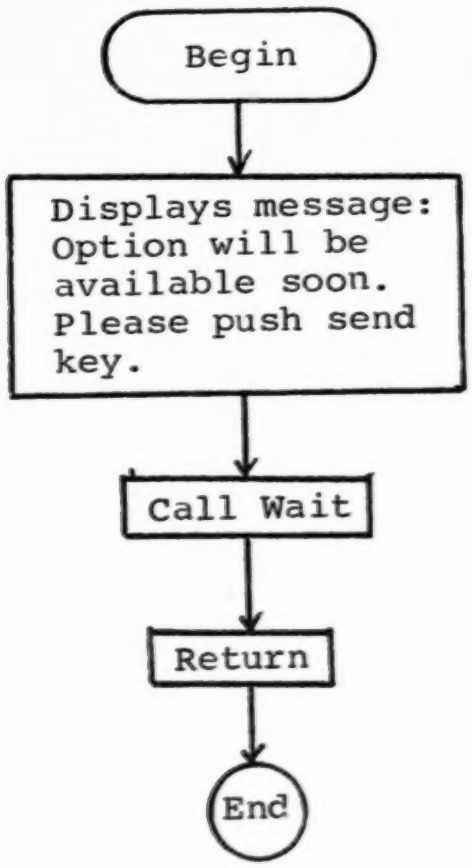


Variables

IREP - memory display buffer

SUBROUTINE UNAVAIL

Displays a message.

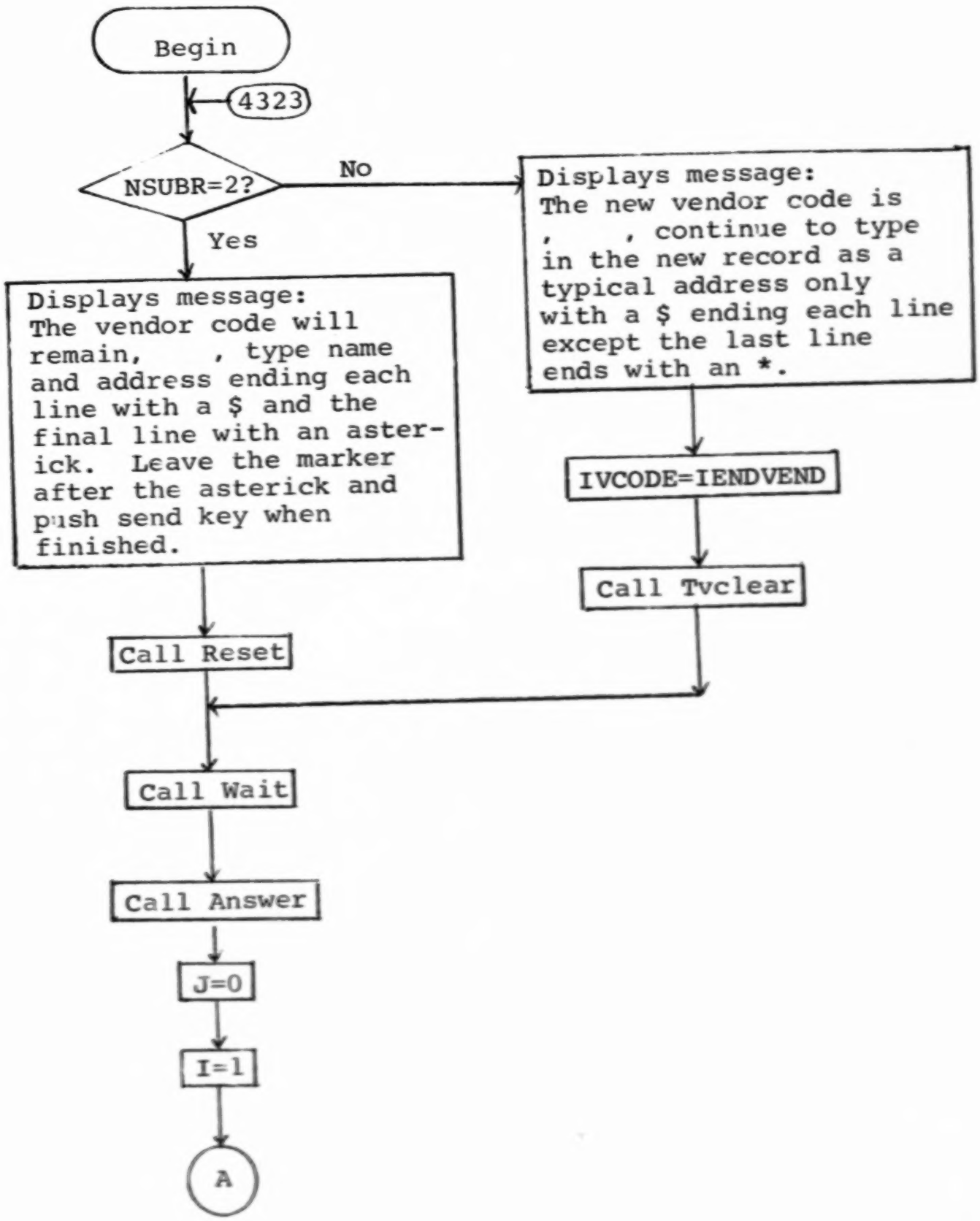


Variable

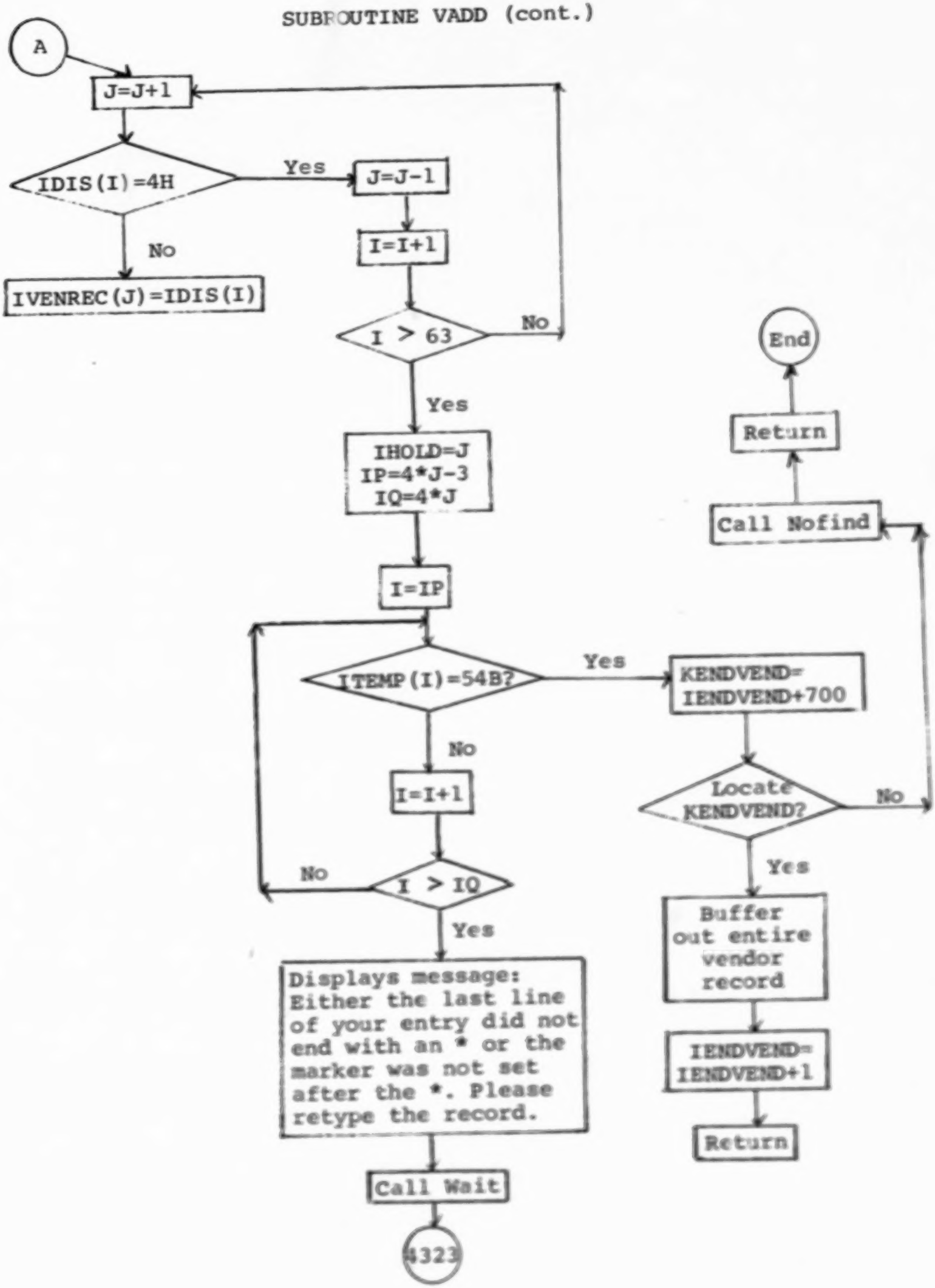
B - memory display buffer

SUBROUTINE VADD

A user types in a new vendor record which is stored in the vendor file. The vendor code corresponding to the record is displayed for the user's information. If the marker is not set after the entered record, the record must be re-entered.



SUBROUTINE VADD (cont.)

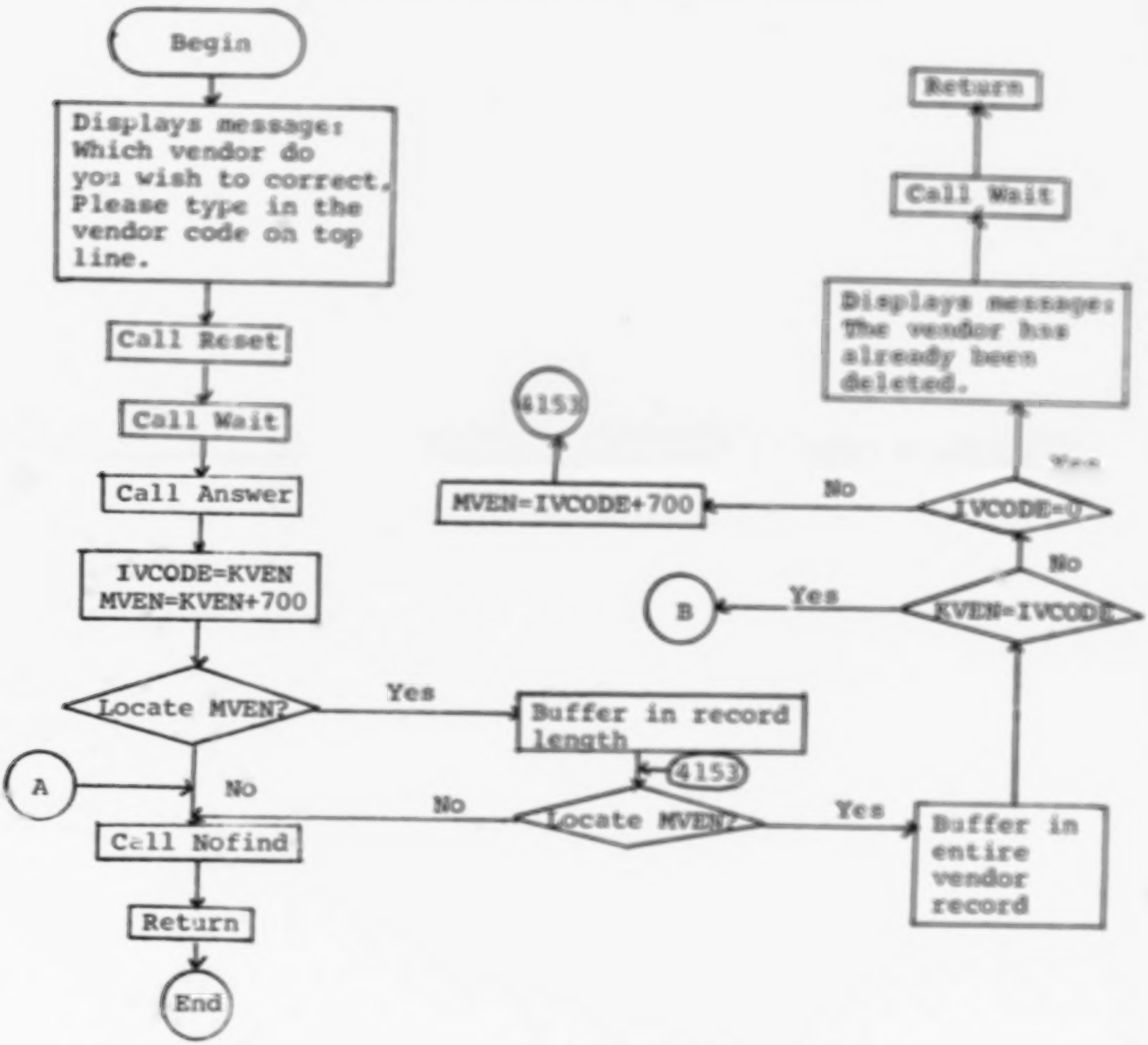


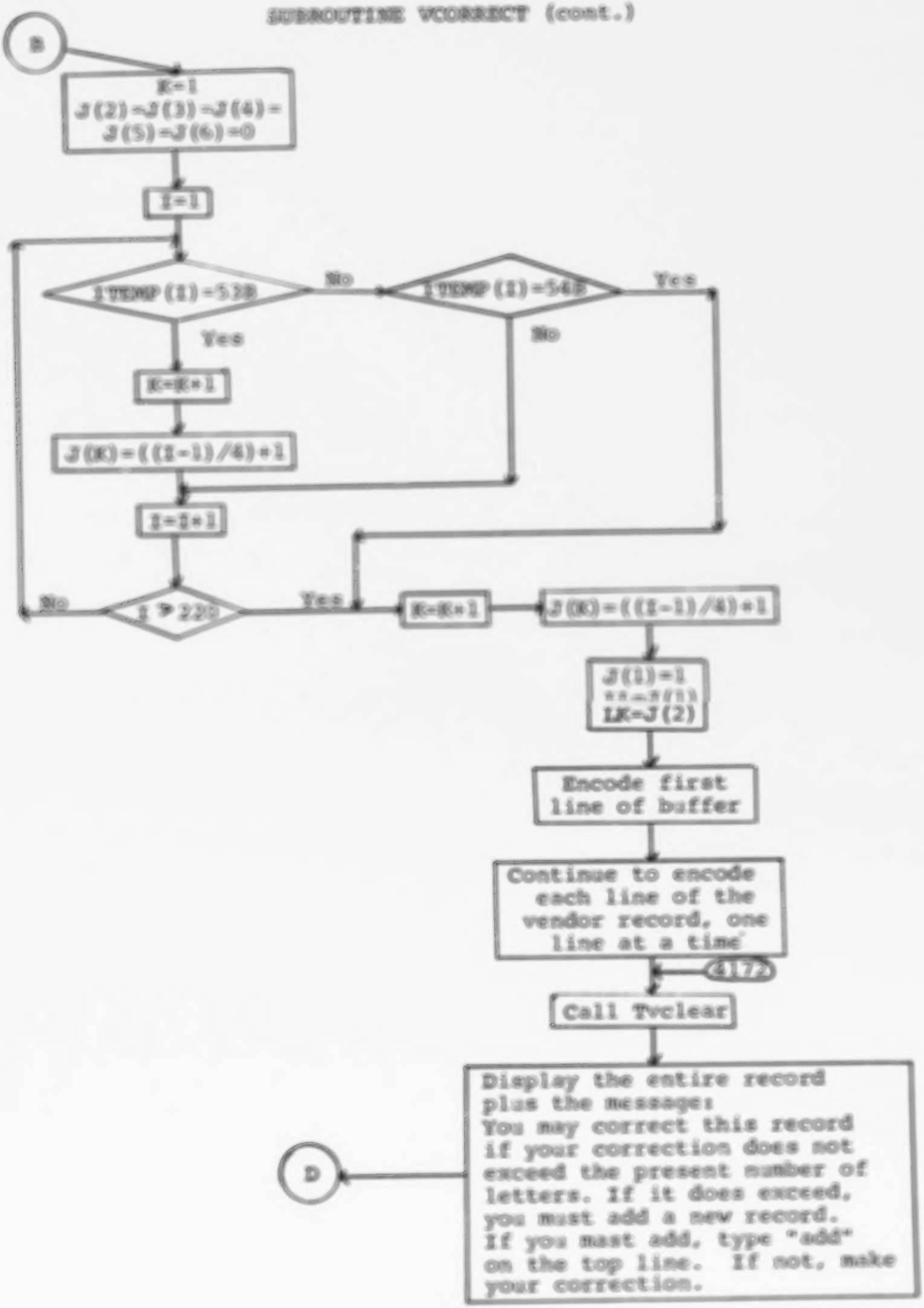
Variables

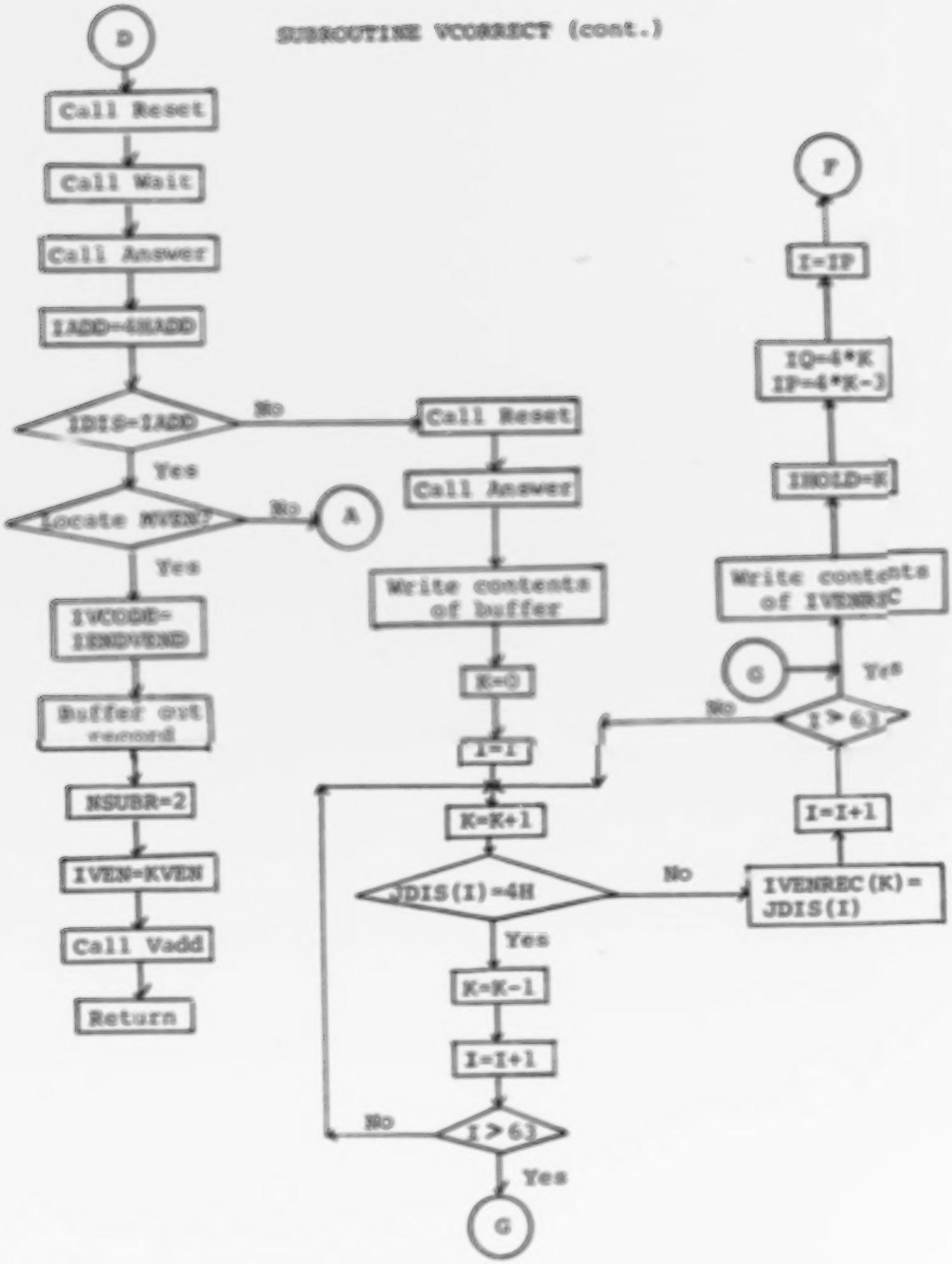
NSUBR - variable specifying whether the new record is entirely new or is an added record because of corrections.
IENDVEND - vendor code of entirely new record.
IVCODE - vendor code on memory record.
JDIS - memory display buffer
IVEN - vendor code of corrected record.
J - counter
IVENREC(J) - array containing vendor record.
IHOLD - length of record.
IP, IQ - variables used for eliminating blanks in typed-in record before it is stored.
ITEMP(I) - equivalent to a character of the vendor record.
KENDVEND - location of added-on record
54B - octal code for *.

SUBROUTINE VCORRECT

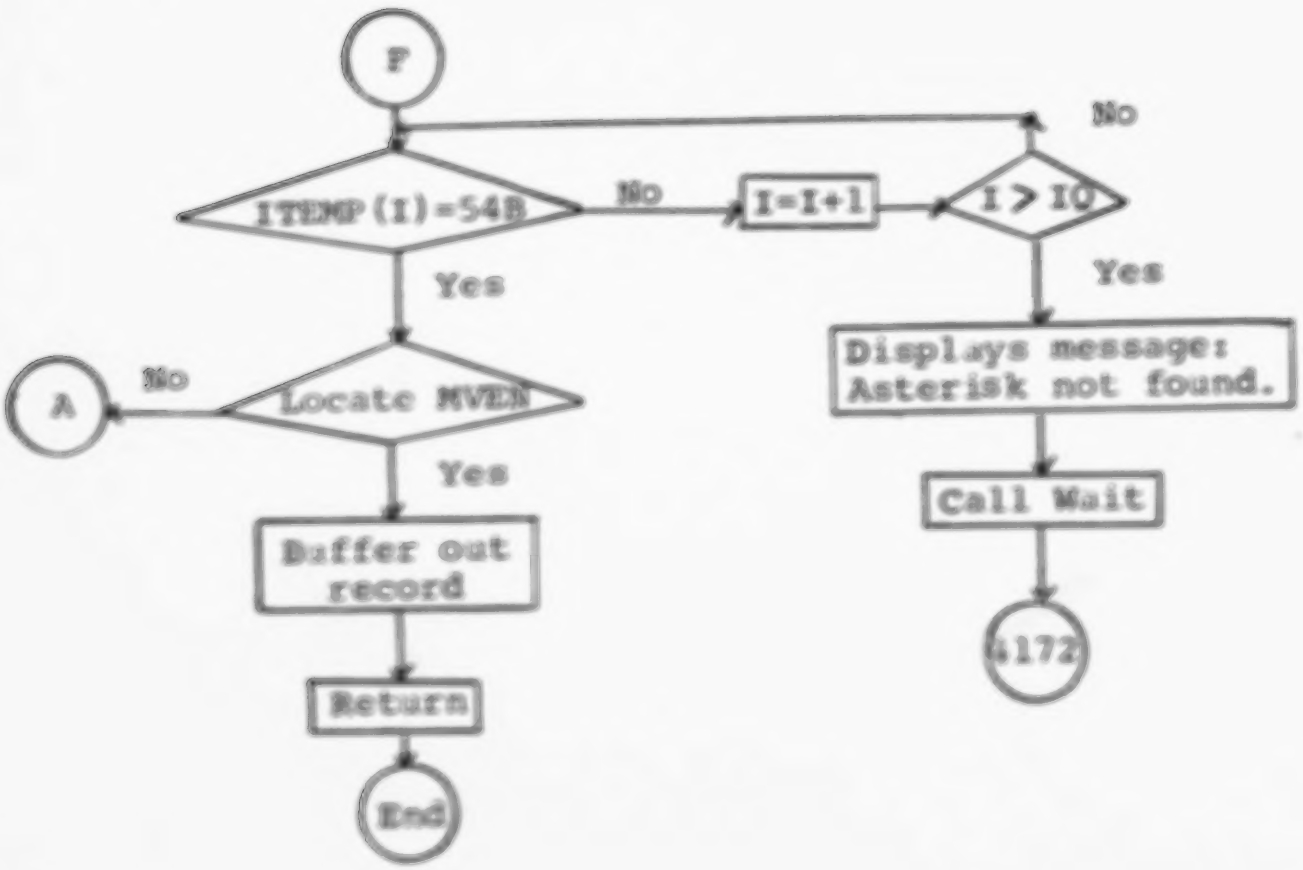
Searches for a particular vendor record (by vendor code) specified by the user. If the record has previously been deleted, a message is displayed stating so. Once the record is found, it is displayed with a message saying the user has two alternatives: 1) to correct the record if his correction does not exceed the present number of characters, or 2) to delete this record and add a new one if his correction goes beyond the available space. Thus, he either corrects this record in place or calls on the VADD subroutine to add a new record.




B
K=1
J(2)=J(3)=J(4)=
J(5)=J(6)=0
I=1
ITEMP(I)=538
No
ITEMP(I)=548
Yes
Yes
No
K=K+1
J(K)=((I-1)/4)+1
I=I+1
No
I > 220
Yes
K=K+1
J(K)=((I-1)/4)+1
J(1)=1
**=J(1)
LK=J(2)
Encode first
line of buffer
Continue to encode
each line of the
vendor record, one
line at a time
4172
Call Tvclear
Display the entire record
plus the message:
You may correct this record
if your correction does not
exceed the present number of
letters. If it does exceed,
you must add a new record.
If you must add, type "add"
on the top line. If not, make
your correction.
D


D
Call Reset
Call Wait
Call Answer
IADD=4HADD
IDIS=IADD
No
Yes
Locate MVEN?
No
A
Yes
IVCODE=
IENDVEND
Buffer out
record
NSUBR=2
IVEN=KVEN
Call Vadd
Return
Call Reset
Call Answer
Write contents
of buffer
K=0
I=1
K=K+1
JDIS(I)=4H
No
Yes
K=K-1
I=I+1
I>63
No
Yes
G
F
I=IP
IQ=4*K
IP=4*K-3
IHOLD=K
Write contents
of IVENREC
G
Yes
I>63
No
I=I+1
IVENREC(K)=
JDIS(I)

SUBROUTINE VCORRECT (cont.)

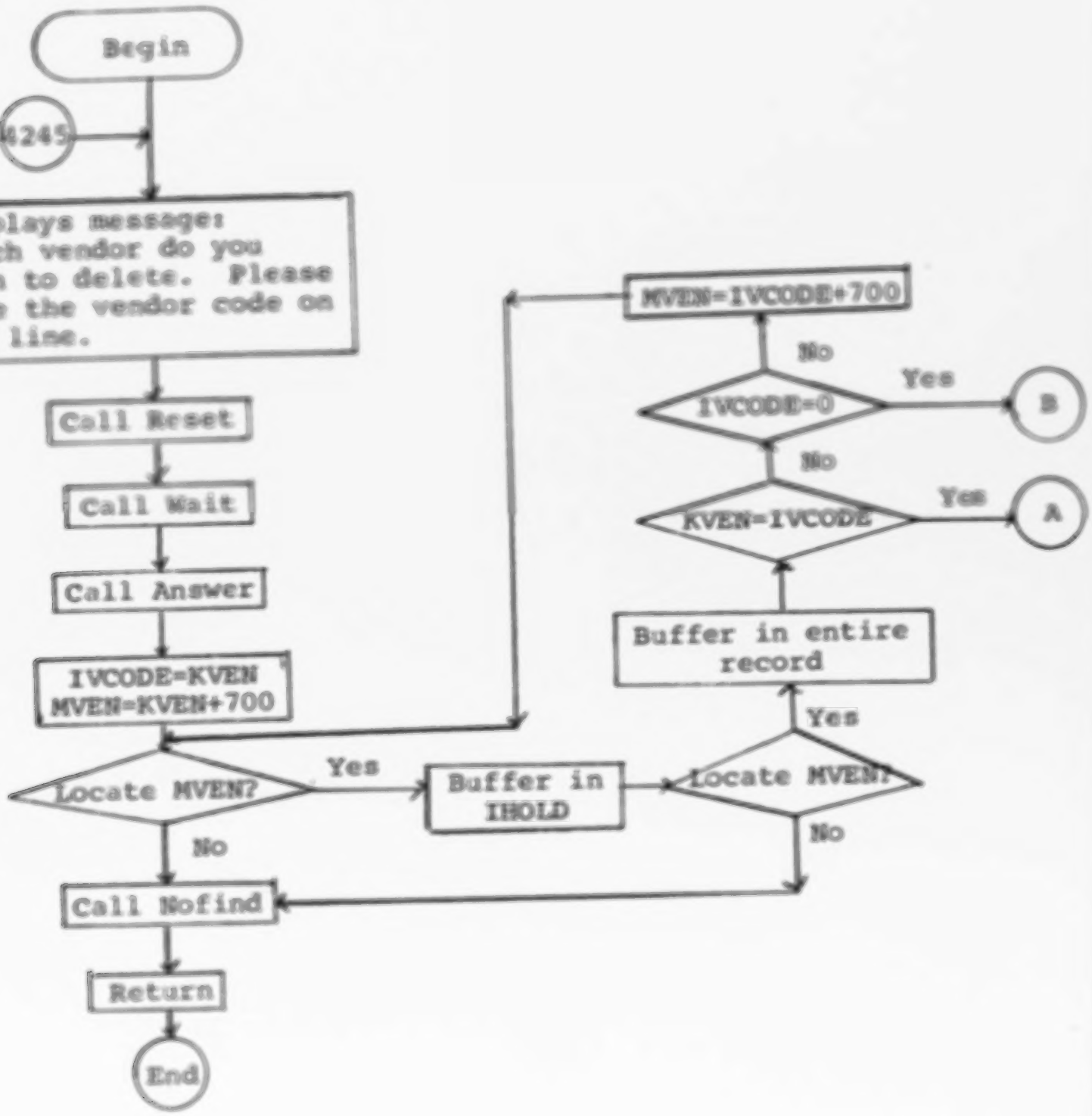


Variables

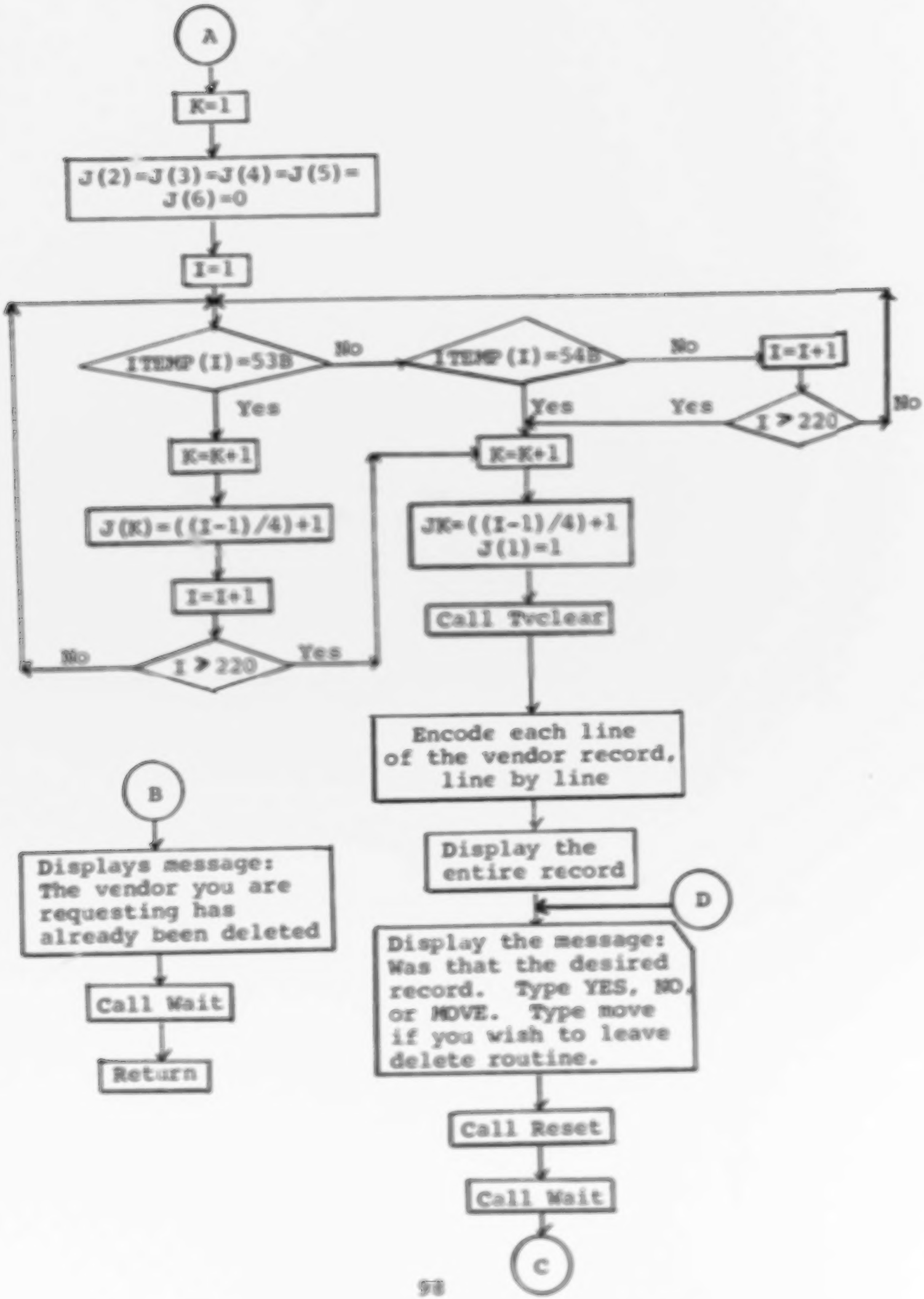
JDIS - memory display buffer
KVEN - vendor code typed in by user
IVCODE - vendor code of record in memory
MVEN - location of vendor record
K - counter
J(1), J(2), J(3), J(4), J(5), J(6) - variables designating the beginning and ending locations of each line of the vendor record.
I - counter
ITEMP(I) - equivalent to a character of the vendor record
53B - octal code for $
54B - octal code for *
ICOUNT - number of characters in a vendor record
IADD - testing variable to see if user wishes to add a new record
IENDVEND - first available space at the end of the vendor file
NSUBR - variable specifying whether the new record is entirely new or is an added record because of corrections.
IVEN - equivalent to KVEN
IVENREC(K) - array containing the vendor record
IP, IQ - variables used for eliminating blanks in typed-in records before storing them

SUBROUTINE VDELETE

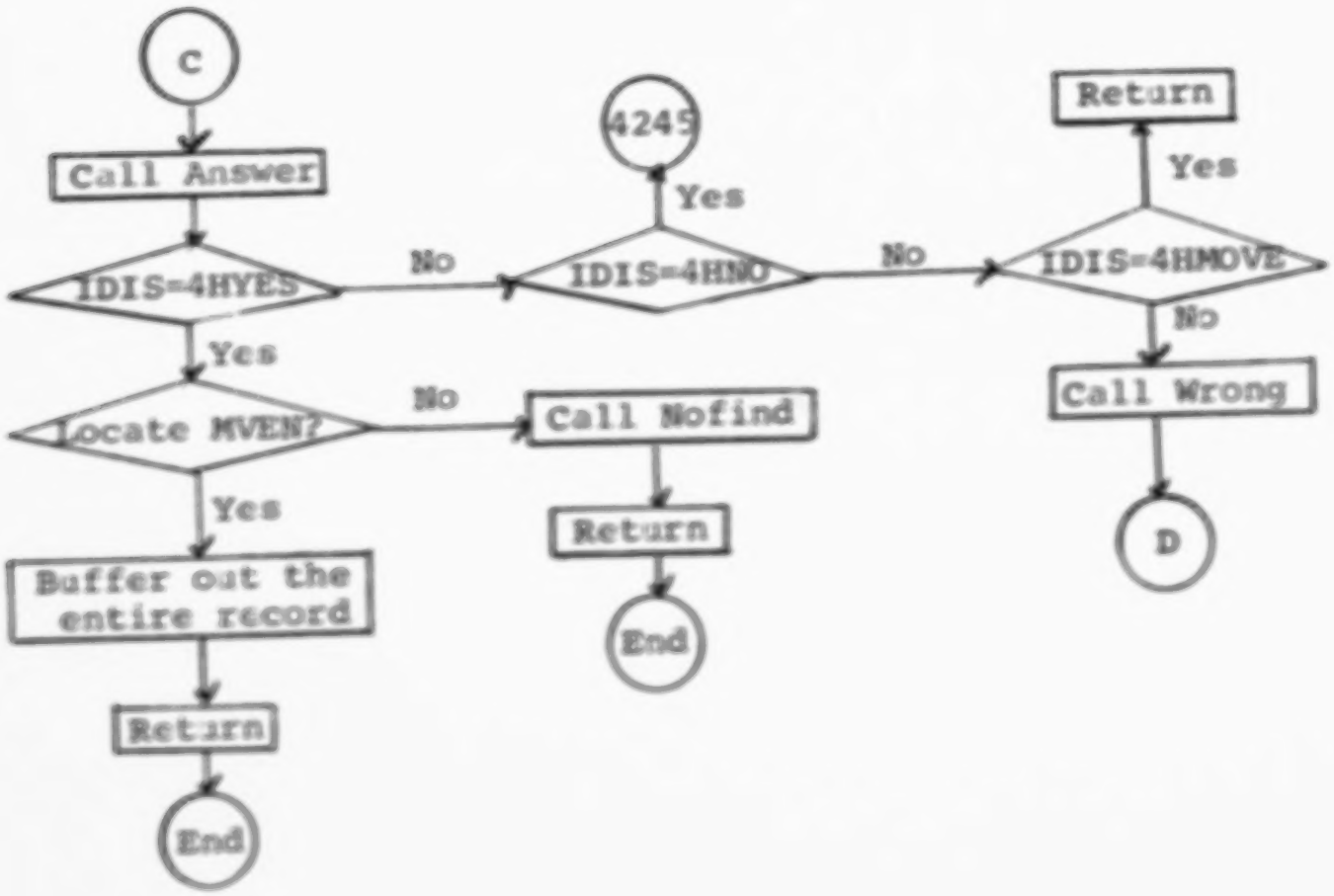
Searches for a particular vendor record (by vendor code) specified by the user. If the record has already been deleted, a message is displayed stating such. Otherwise, the record is displayed along with a message asking whether or not it is the correct record. The user types in YES, NO, or MOVE to show his decision. MOVE will take him out of the delete routine. NO will start the subroutine over with the user typing in a new vendor code. YES places zeros in the vendor code and, thus deleting the record.



SUBROUTINE VDELETE (cont.)



SUBROUTINE VDELETE (cont.)

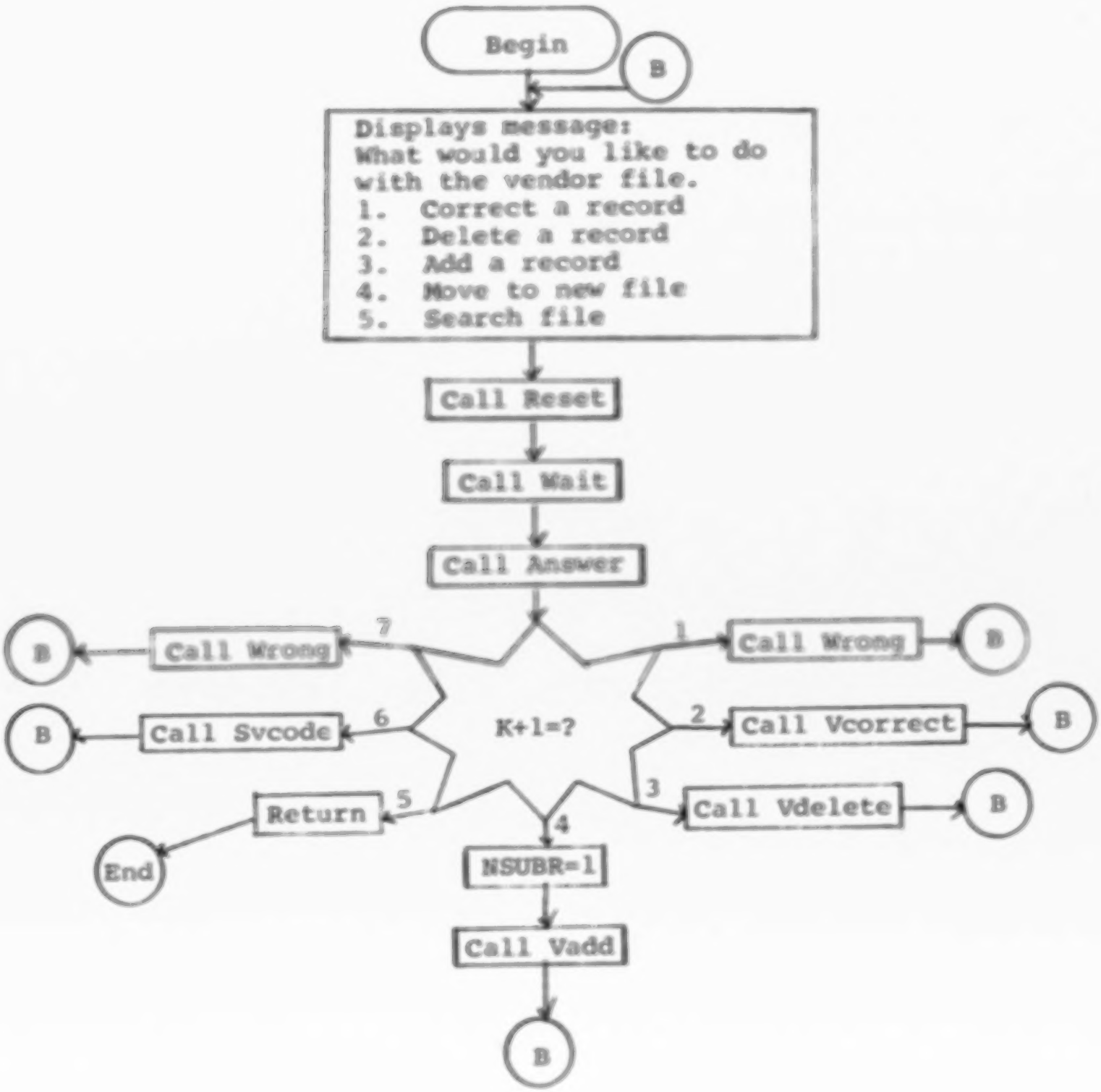


Variables

JDIS - memory display buffer
KVEN - vendor code typed in by user.
IVCODE - vendor code on record in memory.
MVEN - location of desired vendor record in memory
K - counter
J(1),J(2),J(3),J(4),J(5),J(6) - variables designating the beginning and ending locations of each line of the vendor record.
I - counter
ITEMP(I) - equivalent to a character of the vendor record
53B - octal code for $
54B - octal code for *
IDIS - memory storage buffer
4HYES,4HNO,4HMOVE - variables used to test for decision of user

SUBROUTINE VENDFILE

A general calling program for all subroutines involving vendor records.

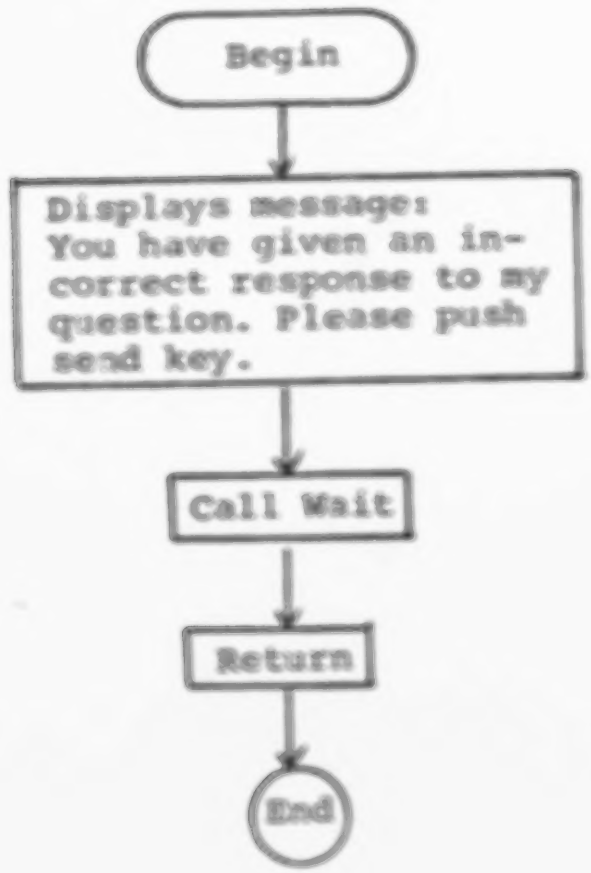


Variables

JDIS - memory display buffer
K - parameter specifying number of choice from message
NSUBR - variable specifying the record to be added is entirely new and not a record that is being corrected.

SUBROUTINE WRONG

Displays a message.



Variables

IRESP - memory display buffer

APPENDIX G

VARIABLE NAME*	DEFINITION*	SUBROUTINES WHERE USED
301	location of first record of new records	SAUTH, SORDER
501	location of first record in purchase file	PSAUTH, PSORDER, SVEND
ALPHA(1)	alphabetical array; when I=1,equals A-block	SAUTH
B	memory display buffer	UNAVAIL
CHO	memory display buffer	SLC, CORECT, PSAUTH, SAUTH, SORDER
IANS	storage buffer	PSAUTH, PURDELT, DELETE, SIGNOFF, SVEND, TOOFAR
IBLANK	check for blanks	CORECT, OSONEW, NEWREC, PURCRDT
IBLOCK	location of desired record	SLC, CORECT, SAUTH, SORDER, PURDELT, RECORD, PURCRDT, DELETE, PSORDER, BACK
ICK	location of flag in memory (used as a testing device)	PSAUTH, SAUTH
IDIS	memory display buffer	SLC, CORECT, PSAUTH, SAUTH, OSONEW, SORDER, PURCRDT, PSORDER, SVEND, OSOFILE(K), PFILE(K)
IDIS(185)	location of middle of order number in the record on the display screen	CORECT, PURCRDT
IERR	memory display buffer	NOFIND
IORNO	order number is displayed records (and in records on file)	CORECT, SORDER, PSORDER

VARIABLE NAME*	DEFINITION*	SUBROUTINES WHERE USED
IPLACE array	location of end record of each alphabetical block. However, IPLACE(1)=1, is the first location in the file	RECORD
IPLACE(28)	location of last record on file in Z's	RECORD
IPLACE(29)=MOUT	location of last record of the file (includes new records)	PSAUTH, SAUTH
IREP	memory display buffer	TOOSMALL, ERROR
IRES	memory display buffer	PURDELT, PURCRDT, SIGNOFF, SETMARK, NOFLAG
IRESP	memory display buffer	WRONG
ISTOP	test for STOP message	PSAUTH
ITEMP(1)	equivalent to first letter of author's name	PSAUTH, SAUTH
ITEMP(I)	equivalent to name of author on the record. However, its length varies according to the number of letters typed by the user.	SAUTH
ITEMP(K)	equivalent to name of author on the record. However, its length varies according to the number of letters typed by the user	PSAUTH
IVEN	vendor code on each record	SVEND
JA=IPLACE(I)+1	location of first record in desired alphabetical block	SAUTH
JA=301	see: 301	SAUTH
JAUTH	equivalent to first letter of author's name on the desired record	CORECT

VARIABLE NAME*	DEFINITION*	SUBROUTINES WHERE USED
JB=IPLACE(29)	see: IPLACE(29)	SAUTH
JB=IPLACE(I+1)	location of last record in desired alphabetical block	SAUTH
JDIS	storage buffer	OSONEW, NEWREC
JDIS(185)	location of middle of order number in the record on the display screen	OSONEW, NEWREC
JI	location of last record on the file	OSONEW
JK	location of last record in purchase file	PSAUTH, NEWREC, PSORDER
JORNO	order number typed in	PSORDER, SORDER
JTEMP(1)	equivalent to first letter of author's name typed by user	CORECT, SAUTH
JTEMP(K)	name of author typed in by user	PSAUTH
JTEMP(LI)	name of author typed in by user	SAUTH
JVEN	memory location of typed-in vendor code	SVEND
K	storage buffer (parameter, transfers number of choice to calling program)	CORECT, OSOFILE(K),PFILE(K)
KORNO	location of typed-in order number	SORDER, PSORDER
KVEN	storage buffer	SVEND
L	parameter used to transfer the yes or no response	SIGNOFF(L)
LCC	storage buffer	SLC
LC(I)	LC number in the record	SLC

VARIABLE NAME*	DEFINITION*	SUBROUTINES WHERE USED
LCC(I)	typed-in LC number	SLC
LD	memory display buffer	RECORD
LI	number of letters before the flag in an author's name	FSAUTH
M	counter, storage buffer	FSAUTH, FILECH(M)
MESS	memory display buffer	OSONEW, NEWREC
MOUT=IPLACE(29)	location of the last record of the new record(s).	SLC, SAUTH, SORDER
MZ=IPLACE(28)	location of last record of the Z-block	SLC, SORDER

* Only a partial listing of all the variables with their respective definitions. Each variable may have more than one purpose depending upon its presence in a certain routine. A more complete listing for each subroutine can be found following the flow-chart of each.